

Naval Research Laboratory

Washington, DC 20375-5320

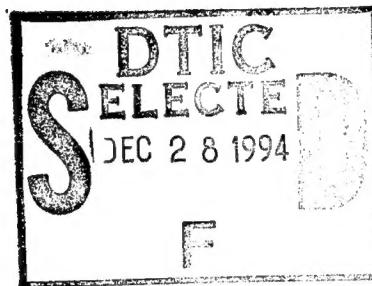


NRL/MR/5360--94-7618

Data Input for RADARPD: Window Version

MOLLY M. YEN

*Airborne Radar Branch
Radar Division*



December 13, 1994

Approved for public release; distribution unlimited.

19941223 063

REPORT DOCUMENTATION PAGE

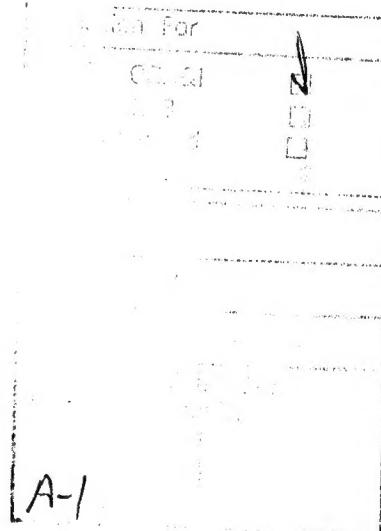
*Form Approved
OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)			2. REPORT DATE		3. REPORT TYPE AND DATES COVERED	
			December 13, 1994			
4. TITLE AND SUBTITLE Data Input for RADARPD: Window Version					5. FUNDING NUMBERS	
6. AUTHOR(S) Molly M. Yen						
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory Washington, DC 20375-5320					8. PERFORMING ORGANIZATION REPORT NUMBER NRL/MR/5360-94-7618	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Air Warfare Center Ad Patuxent River, MD 20670-5304					10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES						
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) An X Window interface inputting data to the RADARPD computer program has been developed. The RADARPD program reads in an input data file and calculates the probability of detection. A brief introduction contrasts the previous VAX based text editor input interface with the new SUN based X Window input interface. After the introduction is a detailed description of the window interface that explains the organization and format of the input window; following that is step-by-step instructions on how to enter and edit data files using the new interface. Concluding the report is the source code.						
14. SUBJECT TERMS					15. NUMBER OF PAGES 58	
					16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL			

CONTENTS

I. INTRODUCTION.....	1
II. WINDOW INTERFACE FORMAT.....	1
III. OPERATION PROCEDURE.....	2
IV. ACKNOWLEDGMENT.....	2
Appendix A.....	A-1
Appendix B.....	B-1



DATA INPUT FOR RADARPD: WINDOW VERSION

I. INTRODUCTION

RADARPD is a computer program which calculates the probability of detection. The program reads in a user inputted data file and then performs the computation. Recently it has been converted from the VAX 11-750 VMS operating system to the SUN Solaris operating system. Previously on the VAX computer, the user had to use a text editor to create or edit the input data file. This input procedure was quite tedious since the data file usually consisted of a long string of numbers, simply separated by commas (Figure 1). The user not only had to remember the order in which to input the data values, but had to remember what each data value represented as well. A window interface for creating and editing the input data file has been developed for the SUN computer, using C language and the NRLx Toolkit, an X Window toolkit. The window interface uses graphical objects such as buttons and labeled text fields. It has not only simplified the input process, but is extremely user friendly as well.

II. WINDOW INTERFACE FORMAT

The window interface consists of a main menu window, Figure 2, and nineteen subwindows, Figure 3 through Figure 21. Each subwindow contains a group of related data values, such as Signal Parameters, Antenna Characteristics, etc. The user is able to open a subwindow or perform certain functions, such as saving the data file and executing the calculation, by simply clicking on a button from the main menu window. Different types of data can be entered, such as integers, floating points, and character strings. There are ten fields on each subwindow, four representing integer values (Item1, Item2, Item3, and Item4), and six representing floating point values (Var1, Var2, Var3, Var4, Var5, and Var6). Each field is labeled with its name and its unit (deg., dB, etc.), if applicable. Not all of the ten fields are used for each subwindow function. For example, only six fields are used for Signal Parameters, Figure 3, while all the fields are used for Antenna characteristics, Figure 8. Only one subwindow, Title for Printout and Plot, Figure 12, contains character string fields.

III. OPERATION PROCEDURE

The user can use this window interface program to create a new data file or edit an existing data file. Since each case has different requirements, the data file does not have a fixed size. The user can click and enter data for as many subwindows as needed. The Execute subwindow, Figure 21, is the only required subwindow for all cases, and should always be the last subwindow to be entered.

The input procedure is as follows:

- (1) Create a new data file:
 - (a) Click on "Open a new data file" button.
 - (b) Click the desired subwindow button on the main menu.
 - (c) Enter the values into the text fields.
 - (d) Save the window when completed by clicking on "Save this new window" button.
 - (e) Repeat b, c, and d as many times as needed.
 - (f) Close the new data file by clicking on the "Close the new data file" button.
- (2) Retrieve and edit an existing data file:
 - (a) Click on "Open an existing data file" button.
 - (b) Click on the desired subwindow button that needs to be edited.
 - (c) The old values will appear in the subwindow, Figure 10. Make the proper changes.
 - (d) Save the retrieved window when completed by clicking on the "Save this in window" button.
 - (e) Repeat b, c, and d as many times as needed.
 - (f) Close the retrieved data file by clicking on the "Close this existing data file" button.

IV. ACKNOWLEDGMENT

I would like to thank Mr. Fred Staudaher who gave me the idea to upgrade the RADARPD input data file to a window interface version.

Appendix A
LIST OF FIGURES

Figure 01	VAX Input Data File.....	A-02
Figure 02	Main Menu	A-03
Figure 03	Signal Parameters.....	A-04
Figure 04	Radar Detection Parameters.....	A-05
Figure 05	AEW Geometry.....	A-06
Figure 06	Target Height Limits.....	A-07
Figure 07	Target Range Limits.....	A-08
Figure 08	Antenna Characteristics.....	A-09
Figure 09	Antenna Data Points.....	A-10
Figure 10	Clutter Characteristics.....	A-11
Figure 11	Cancellation Characteristics.....	A-12
Figure 12	Title for Printout & Plot.....	A-13
Figure 13	Target Velocity Limits.....	A-14
Figure 14	PD Contour Plot Data.....	A-15
Figure 15	Extended Surface Target Parameters.....	A-16
Figure 16	Parameters for Isometric Plot.....	A-17
Figure 17	Antenna Pattern Print/Plot Parameters.....	A-18
Figure 18	Atmospheric Attenuation Parameters.....	A-19
Figure 19	Rain Attenuation Parameters.....	A-20
Figure 20	Jammer Parameters.....	A-21
Figure 21	Execution.....	A-22

```
-----  
2,,3000,,,1.,5.,5.66,732.6,20.,  
3,1,1,4,2,,229.,6.4,30.,,  
4,,,,2500.,,,,  
5,,,,10.,50.,2.,,,  
6,,,,20.,200.,2.,,,  
7,,,,3.,,34.,,18.,,,45.  
9,1,,,1,-43.8,0.8,0.2,10.,,  
112.,,,,  
Test Run  
May 30, 1994  
1623,8,,,40.,,0.2,5.,,,  
1,,,,10.,,,,  
-----
```

Figure 1 Old VAX Input Data File

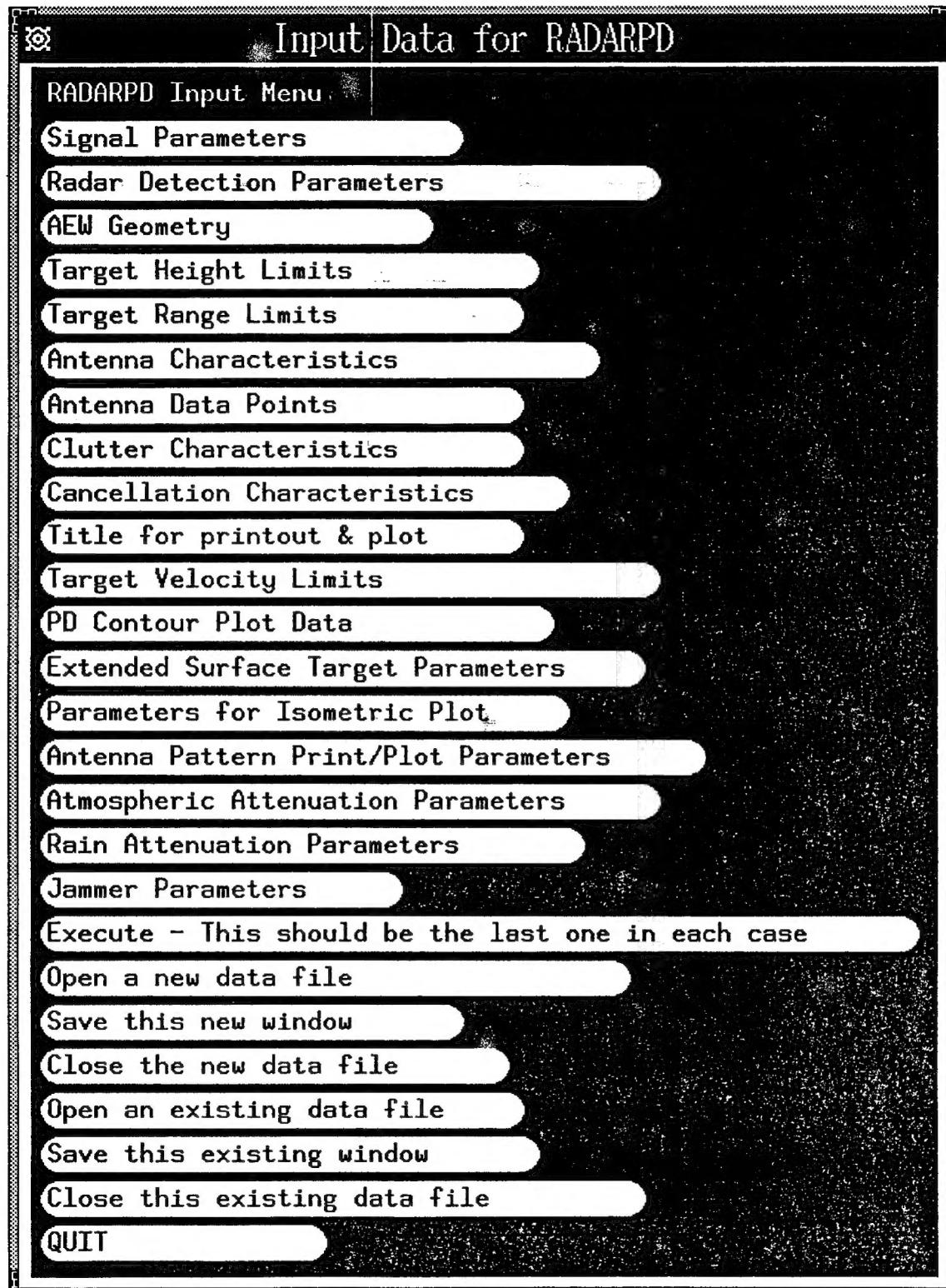


Figure 2 Main Menu

Signal Parameters

Item 1
(S) >
RF Transmit Frequency (MHz)
(S) >
Item 3
(S) >
Item 4
(S) >
Peak Power (Mw)
(S) >
Instantaneous Bandwidth (MHz)
(S) >
Cumulative System Losses (dB)
(S) >
System Noise Temp. (K)
(S) >
Transmit Pulse Width (micro sec)
(S) >
Var6
(S) >

Figure 3 Signal Parameters

Radar Detection Parameters

0-Conolly's law, 1-Marcum/Swerling, 2-Shotland
⊗ >

0 - Fixed, # - Swerling Case
⊗ >

- Independent Groups
⊗ >

- Dependent Pulses in a Group
⊗ >

Var1
⊗ >

Coherent Integration Gain
⊗ >

Probability of False Alarm (-10*log(PFA))
⊗ >

Target Cross Section (sq. meters)
⊗ >

Detection Sensitivity (dB)
⊗ >

Detection Slope (default=1)
⊗ >

Figure 4 Radar Detection Parameters

AEW Geometry

Item 1

(>)

Item 2

(>)

Item 3

(>)

Item 4

(>)

Platform Altitude (ft.)

(>)

Platform Pitch Angle (deg.)

(>)

Platform Roll Angle (deg.)

(>)

Var 4

(>)

Var 5

(>)

Var 6

(>)

Figure 5 AEW Geometry

Target Height Limits

Item 1
(<input type="checkbox"/> >)
Item 2
(<input type="checkbox"/> >)
Item 3
(<input type="checkbox"/> >)
Item 4
(<input type="checkbox"/> >)
Minimum Target Altitude (ft.)
(<input type="checkbox"/> >)
Maximum Target Altitude (ft.)
(<input type="checkbox"/> >)
Increment Altitude (ft.)
(<input type="checkbox"/> >)
Var4
(<input type="checkbox"/> >)
Var5
(<input type="checkbox"/> >)
Var6
(<input type="checkbox"/> >)

Figure 6 Target Height Limits

Target Range Limits	
Item 1	
Item 2	
Item 3	
Item 4	
Minimum Target Range (nmi)	
Maximum Target Range (nmi)	
Increment Range (nmi)	
Var 4	
Var 5	
Var 6	

Figure 7 Target Range Limits

Antenna Characteristics

- of Antenna Pattern Data Points
<input type="button" value="S >"/>
Reference Count (10*full scale in dB)
<input type="button" value="S >"/>
0 – Horizontal Polarity, 1 – Vertical Polarity
<input type="button" value="S >"/>
0-T/R parameter,1-Transmit pattern,2-Receive pattern,3-VPAT routine
<input type="button" value="S >"/>
Antenna az. pointing angle (deg.)
<input type="button" value="S >"/>
Antenna Reference Gain (dB)
<input type="button" value="S >"/>
Min Angle of Antenna Pattern(if II!=0), Vertical Beamwidth (deg)
<input type="button" value="S >"/>
Increment of Pattern (if II != 0)
<input type="button" value="S >"/>
Full Scale of Antenna Chart (dB)(if II!=0), Elevation offset Angle (deg)
<input type="button" value="S >"/>
Vertical Pattern Phase Parameter
<input type="button" value="S >"/>

Figure 8 Antenna Characteristics

Stored Antenna Data Points

- of Current Antenna Point
Recorder Count (10*value in dB)
Item 3
Item 4
Voltage Pattern Sign (0, +1, -1)
Var2
Var3
Var4
Var5
Var6

Figure 9 Antenna Data Points

Clutter Characteristics

0-Land,1-Sea,Model Clutter Coeff func:2/3(L/S)-range, 4/5(L/S)-grazing

>1

Diffration Option: 0-Interpolate,1-Multi Path,2-Fock

>0

0-(all cases except 2/3),Index# of Range Interval for coeff

>0

0-No Forward Scatter,1-Forward Reflection,Forward Power:2-Min,3-Max

>1

Clutter Reflection Coeff (dB).(for cases 4/5 coeff taken at 10 (deg))

>-43.799999

Horizontal Beamwider (deg.)

>0.800000

Pulse Width (post-compressed) (micro sec.)

>0.200000

Wave Ht(ft):0-0',1-<1',2-(1-3)',3-(3-5)',4-(5-8)',5-(8-12)',6-(12-20)'

>5.000000

Min range (cases 2/3)(nmi), Slope of Coeff(case 4/5) (dB/decade)

>0.000000

Increment Range (cases 2/3) (nmi)

>0.000000

Figure 10 Clutter Characteristics

Cancellation Characteristics

0-Single Cancellation ratio, # - Cancellation ratio for #th range value
⊗ >

- Pulses Per Antenna Beamwidth
⊗ >

0-No Loss func., 1-Single Delay Canceller, 2-Double Delay Canceller
⊗ >

Item 4

⊗ >

Cancellation Ratio (dB)
⊗ >

Threshold Loss Coefficient - A1 (if unknown use 0)
⊗ >

Threshold Loss Coefficient - A2 (if unknown use 5)
⊗ >

Threshold Loss Coefficient - B1 (if unknown use 1)
⊗ >

Threshold Loss Coefficient - B2
⊗ >

Threshold Loss Coefficient - B3
⊗ >

Figure 11 Cancellation Characteristics

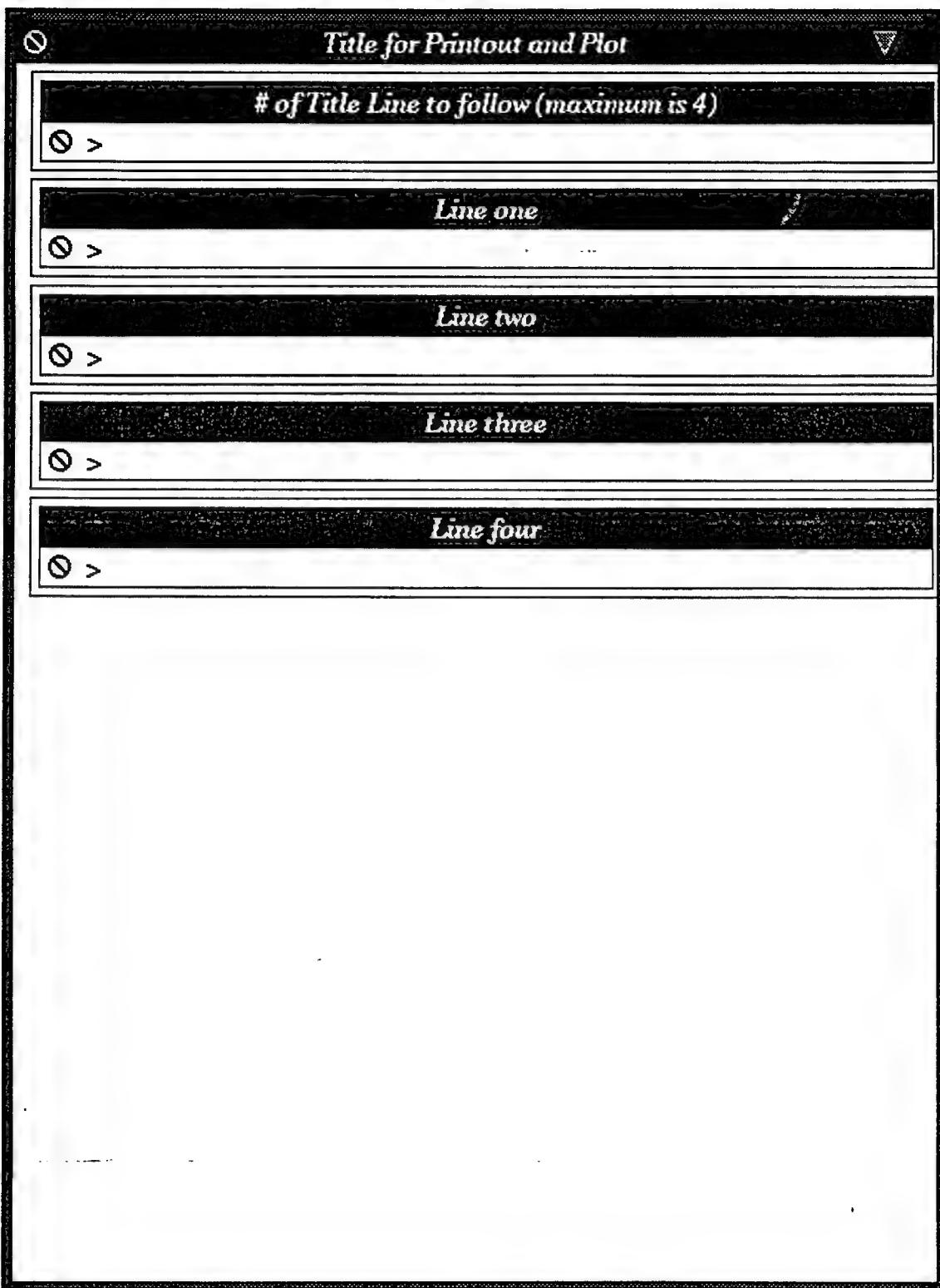


Figure 12 Title for Printout & Plot

Target Velocity Limits

0-For Regular PD Calc, 1-For Cumulative PD Calc

Item 2

Item 3

Item 4

Minimum Target Range Rate (ft/sec)

Maximum Target Range Rate (ft/sec)

Increment Rate (ft/sec)

Scan Interval (sec)

Range Increment Between Printouts (nmi)

Cumulative Prob Level for Special Print

Q >

V >

Figure 13 Target Velocity Limits

PD Contour Plot Data

0-Regular Plots only, 1-Automatic Contour Plots
S >

of Contour Levels (l+1/step_size)
S >

Item 3
S >

Item 4
S >

Maximum Target Altitude (ft.)
S >

Range Scale (nmi/in.)
S >

Height Scale (ft/in.)
S >

Var4
S >

Var5
S >

Var6
S >

Figure 14 PD Contour Plot Data

Extended Surface Target Parameters

0 - Header, 1 - Detail

0 - Point Model, 1 - Area Model, # of point (if II=1)

Total # of detail points (detail cards must follow)

Item 4

Height (ft.)

Area (sq. ft.)

Var3

Var4

Var5

Var6

Figure 15 Extended Surface Target Parameters

Parameters for Isometric Plot		
<input type="radio"/> Item 1: 0 - None (1,2,3)	<input type="radio"/> 1 - Signal/Noise	
<input type="radio"/> 2 - Signal/(Clutter + Noise)	<input type="radio"/> 3 - Probability of Detection	
Height Independent Plots : 10 - Reset to zero(11,12,13)		
<input type="radio"/> 11 - sh1	<input type="radio"/> 12 - sh2	<input type="radio"/> 13 - sh3
Height Dependent Plots: 20 - Reset to zero(21,22,23)		
<input type="radio"/> 21 - sh1	<input type="radio"/> 22 - sh2	<input type="radio"/> 23 - sh3
<input type="radio"/> Item 2: 0 - Reset	Height Independent Plots: 1 - Clutter/Noise	
<input type="radio"/> 2 - Threshold Loss	<input type="radio"/> 3 - Cancellation Ratio	
Height Dependent Plots: 1 - Clutter/Noise		<input type="radio"/> 2 - Signal/Noise
<input type="radio"/> 4 - Signal/(Clutter+Noise)		<input type="radio"/> 8 - Prob. of Detection
Item 3		
<input type="radio"/> >		
Item 4		
<input type="radio"/> >		
Range Scale (nmi/in.)		
<input type="radio"/> >		
dB or Pd scale (dB/in. or unit/in.)		
<input type="radio"/> >		
Minimum dB or Pd (dB or unit)		
<input type="radio"/> >		
Maximum Z Dimension (in.)		
<input type="radio"/> >		
Height Scale (ft/in.)		
<input type="radio"/> >		
Fixed XMAX for Plotting (nmi)		
<input type="radio"/> >		

Figure 16 Parameters for Isometric Plot

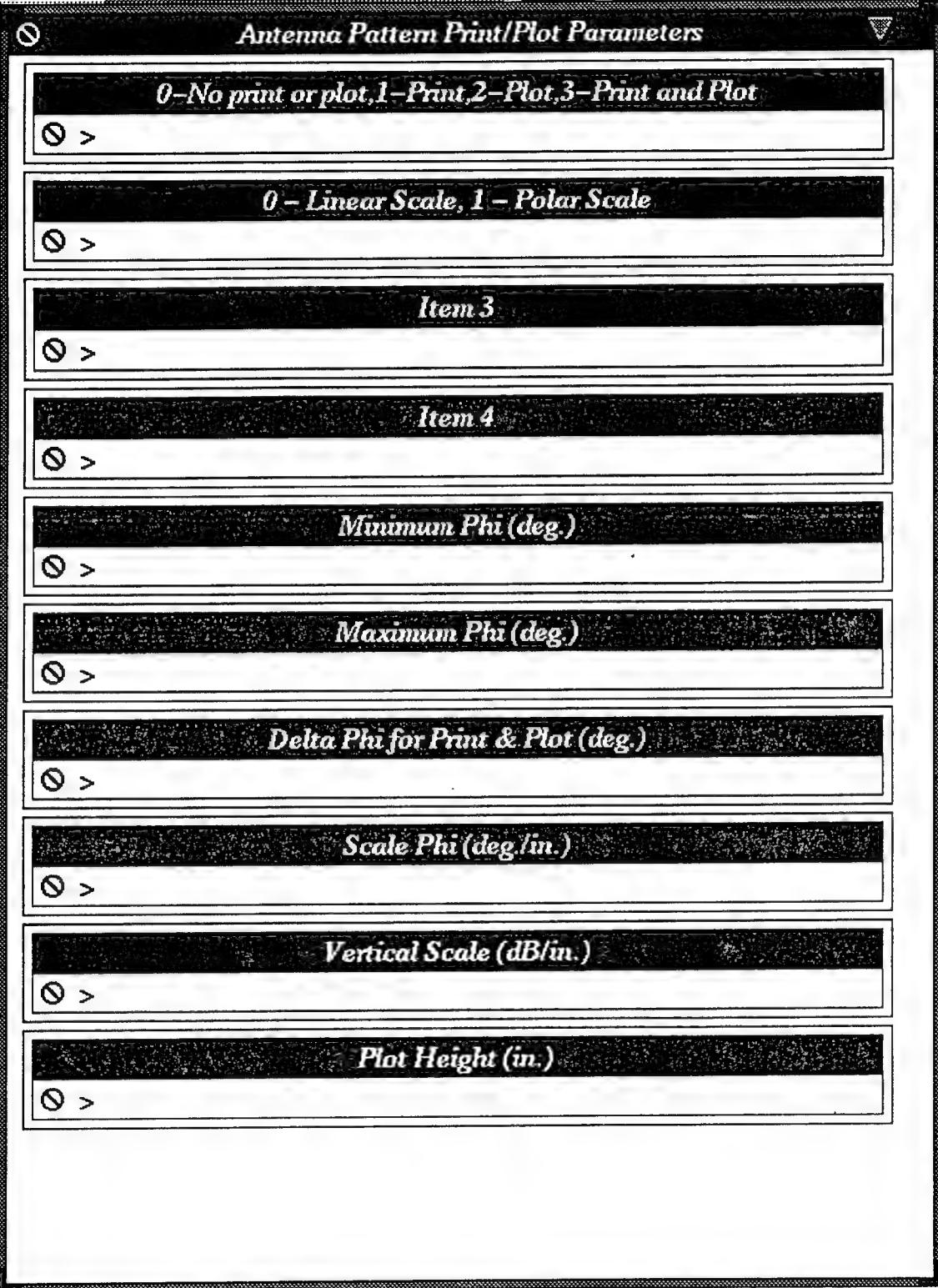


Figure 17 Antenna Pattern Print/Plot Parameters



Figure 18 Atmospheric Attenuation Parameters

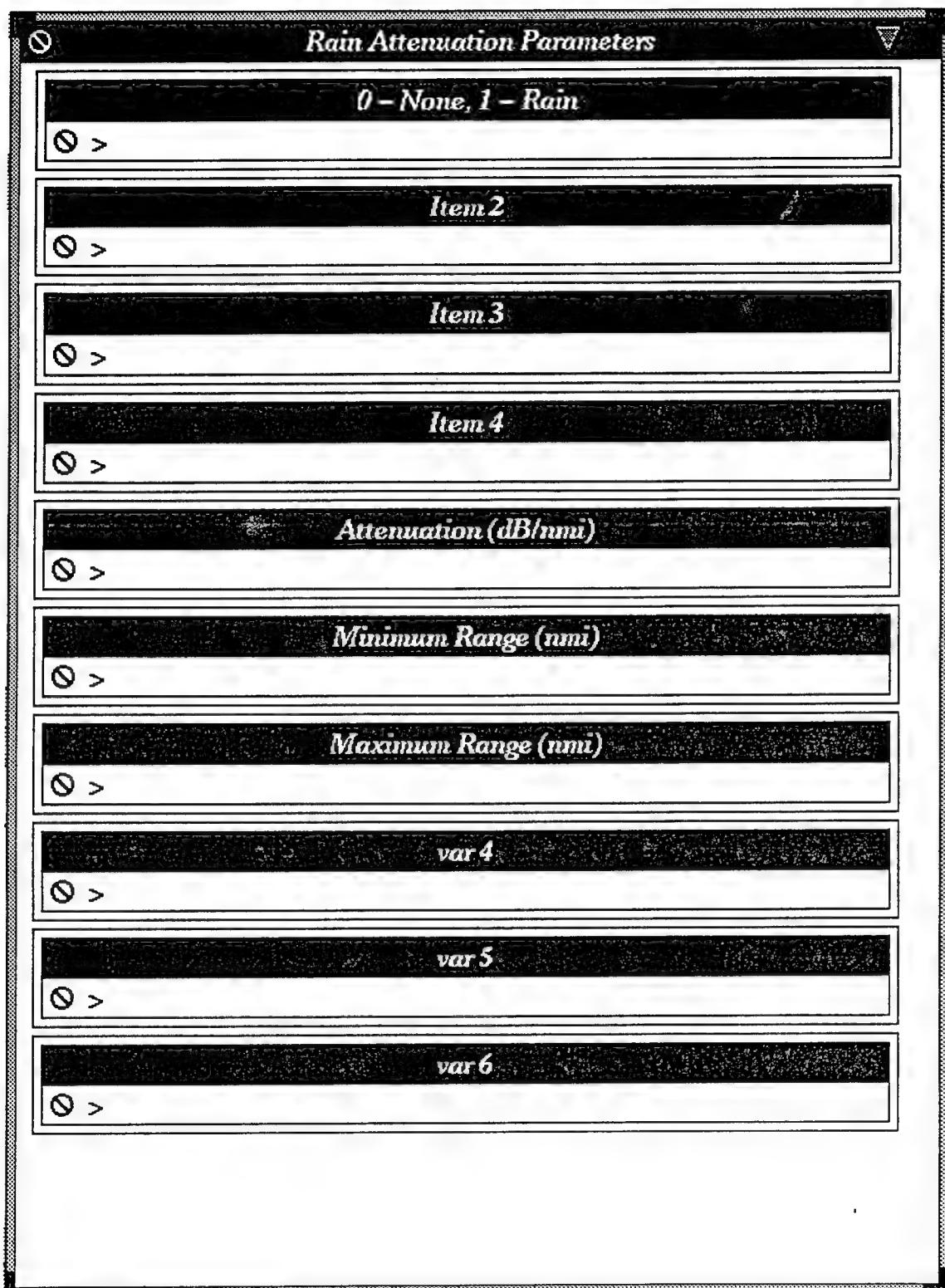


Figure 19 Rain Attenuation Parameters

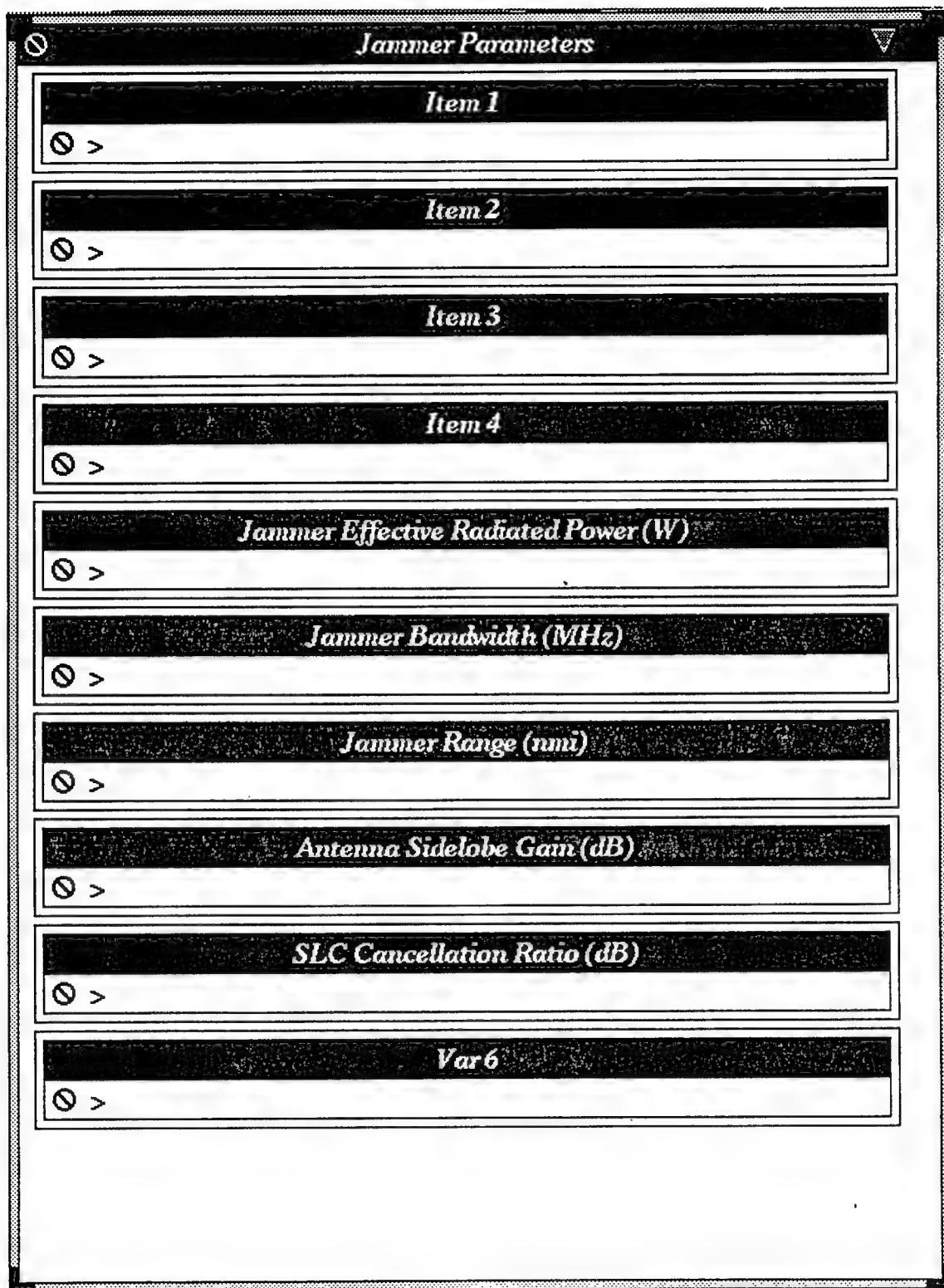


Figure 20 Jammer Parameters

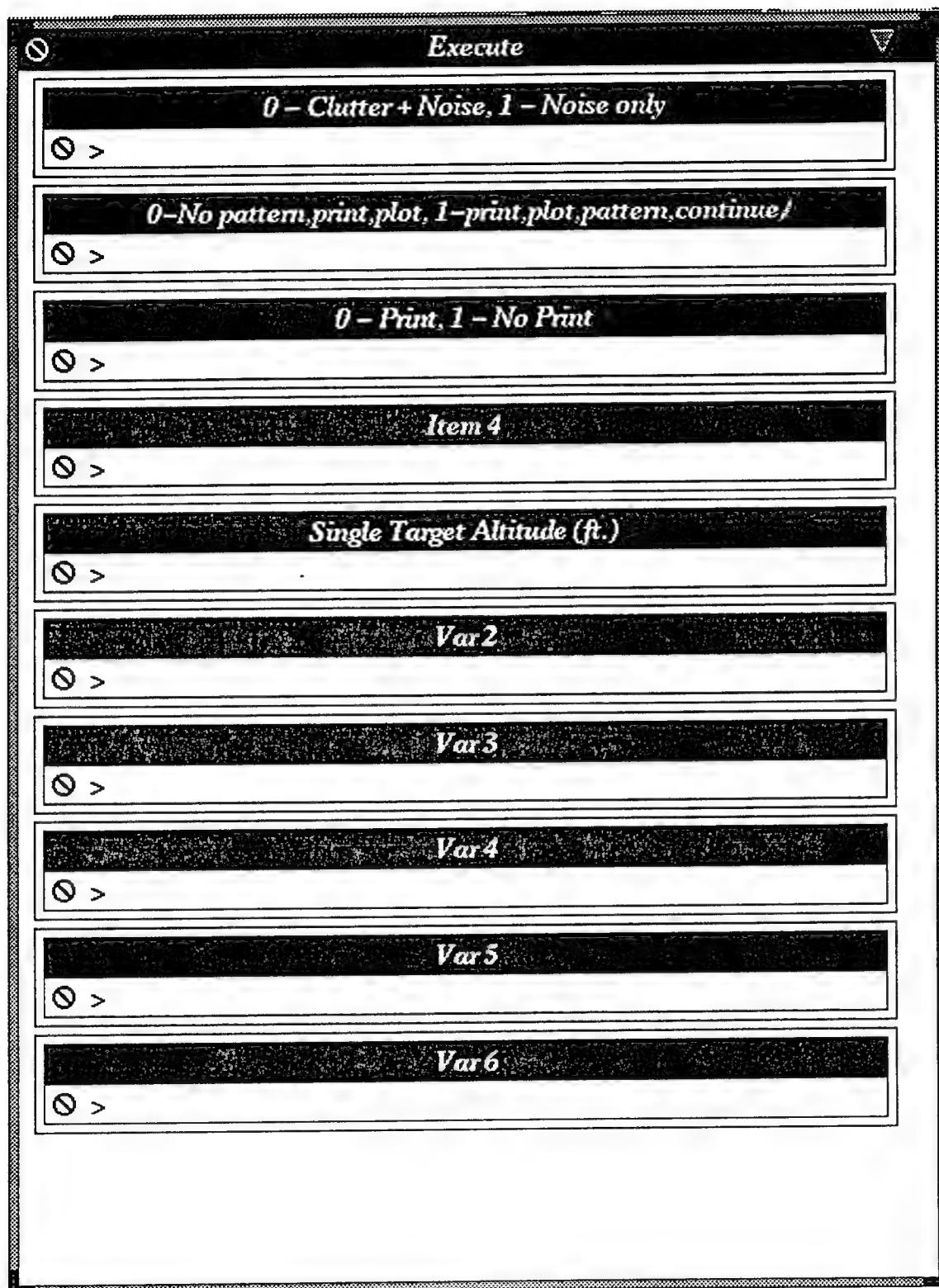


Figure 21 Execution

Appendix B

SOURCE CODE LISTINGS

```
#ifndef INDATA_H
#define INDATA_H
#include <NRlx.h>
#include <NRlxUtil.h>
#endif
```

```

/*
 * This program uses NRLX Toolkit to create a window interface */
/*
 * Version of RADARD Input data file */
/*
 * M. Yen Aug. 1993 */
/*
 * Revision: M. Yen Sept., 1994 */
/*
 * Recompiled and relinked the program with the upgraded NRLX library */
/*
 * Routines. */

#include "indata.h"

struct nrlx
    *console;

struct nrlx
    *popup2, *popup3, *popup4, *popup5, *popup6, *popup7,
    *popup8, *popup9, *popup10, *popup11, *popup12, *popup13, *popup14,
    *popup15, *popup16, *popup17, *popup18, *popup19, *popup20, *popup21,
    *popup22;

struct nrlx
    *menu1, *entry_1, *entry_2, *entry_3, *entry_4, *entry_5, *entry_6,
    *entry_7, *entry_8, *entry_9, *entry_10, *entry_11, *entry_12,
    *entry_13, *entry_14, *entry_15, *entry_16, *entry_17,
    *entry_18, *entry_19, *entry_20;

struct nrlx
    *entry_31,
    *entry_32;
    *entry_33;

struct nrlx
    *entry_40, *entry_41, *entry_42;

struct nrlx
    *text, *text1, *text2, *text3, *text4, *text5,
    *text6, *text7, *text8, *text9,
    *text20, *text21, *text22, *text23,
    *text10, *text11, *text12, *text13, *text14, *text204;

struct nrlx
    *text15, *text16, *text17,
    *text205, *text206, *text207, *text208, *text209,
    *text210, *text211;

struct nrlx
    *text18, *text19, *text20,
    *text212, *text213, *text214, *text215,
    *text216, *text217, *text218;

struct nrlx
    *text21, *text22, *text23,
    *text219, *text220, *text221, *text222,
    *text223, *text224, *text225;

struct nrlx
    *text24, *text25, *text26, *text27,
    *text28,
    *text34, *text35, *text36,
    *text29, *text30, *text31, *text32, *text33,
    *text231, *text232;

struct nrlx
    *text37, *text38, *text39, *text40, *text41, *text42,
    *text43, *text44, *text45, *text46;

struct nrlx
    *text47, *text48, *text49, *text50, *text51, *text52,
    *text53, *text54, *text55, *text233;
}

struct nrlx
    *text56, *text57, *text58, *text59, *text60, *text61,
    *text62, *text234, *text235, *text236;
    *text63, *text64, *text65, *text66, *text67,
    *text237, *text238, *text239, *text240, *text241;
    *text68, *text69, *text70, *text71, *text72,
    *text242, *text243, *text244, *text245, *text246;

struct nrlx
    *text73, *text74, *text75, *text76, *text77, *text78,
    *text80, *text81, *text82, *text83, *text84, *text85,
    *text86,
    *text87, *text88, *text89, *text90, *text91, *text92, *text93,
    *text94, *text95, *text96, *text97, *text98,
    *text247, *text248;

struct nrlx
    *text100, *text101, *text102, *text103, *text104, *text105,
    *text106, *text107, *text245, *text254, *text255,
    *text110, *text111, *text112, *text113, *text114, *text115,
    *text116, *text117, *text118, *text253;
    *text120, *text121, *text122, *text123, *text252, *text253,
    *text254, *text255, *text256, *text257;

struct nrlx
    *text130, *text131, *text132, *text133, *text134,
    *text258, *text259, *text260, *text261, *text262;

struct nrlx
    *text140, *text141, *text142, *text143, *text263,
    *text264, *text265, *text266, *text267, *text268;

struct nrlx
    *text150, *text151, *text152, *text153, *text154;

int ycount, inc_xcount, yycount;
int cardno, loc1, loc2, loc3, cndt;
float v1, v2, v3, v4, v5, v6;
int i1, i2, i3, i4;
int line1, line2, line3, line4;

char *str1, *str2, *str3, *str4;
char $il, $i2, $i3, $i4, $sv1, *sv2, *sv3, *sv4, *sv5, *sv6;
char rstr1[79], rstr2[79], rstr3[79], rstr4[79];
char astr1[79], astr2[79], astr3[79], astr4[79];

void stoi(), stoi3(), stoi4();
void stof1(), stof2(), stof3(), stof4(), stof5(), stof6();
void item6(), item2(), string1(), string2(), string3(), string4();

FILE *fp, *fp1;
char fname[30], infname[30];

void main (argc, argv)
    int argc;
    char **argv;
{
    char quit ();
}

```

```

void signal();
void radardet();
void aewgeo();
void tarhgt();
void tarrng();
void antcha();
void contur();
void surtar();
void isompt();
void anpat();
void atmattn();
void rainatt();
void jammer();
void execute();
void savewin();
void penfile();
void closefile();
void saveinf();
void openinfile();
void closeinfile();
str1=calloc(80,sizeof(char));
str2=calloc(80,sizeof(char));
str3=calloc(80,sizeof(char));
str4=calloc(80,sizeof(char));
sil=calloc(80,sizeof(char));
si2=calloc(80,sizeof(char));
si3=calloc(80,sizeof(char));
si4=calloc(80,sizeof(char));
sv1=calloc(80,sizeof(char));
sv2=calloc(80,sizeof(char));
sv3=calloc(80,sizeof(char));
sv4=calloc(80,sizeof(char));
sv5=calloc(80,sizeof(char));
sv6=calloc(80,sizeof(char));
console = NRLx (MainWindow, NULL,
XPos,10, YPos,10,
Label, "Input Data for RADARD",
Width, 500,
Height, 800,
Background, "white",
FrameColor, "turquoise",
NULL);

menu = NRLx (Menu, console, xPos,10, yPos,10,
Label, "RADARD Input Menu
Plined, TRUE,
TextFont, "10x20",
NULL);

entry_1 = NRLx (Entry, menu,
Label, "signal Parameters
",
```

Proc, signal, "black",
Foreground, "white",
Shape, oval,
NULL);

entry_2 = NRLx (Entry, menu,
Label, "Radar Detection Parameters
Proc, radardet,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_3 = NRLx (Entry, menu,
Label, "AEW Geometry
Proc, aewgeo,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_4 = NRLx (Entry, menu,
Label, "Target Height Limits
Proc, targht,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_5 = NRLx (Entry, menu,
Label, "Target Range Limits
Proc, tarring,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_6 = NRLx (Entry, menu,
Label, "Antenna Characteristics
Proc, antcha,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_7 = NRLx (Entry, menu,
Label, "Antenna Data Points
Proc, antdata,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_8 = NRLx (Entry, menu,
Label, "Clutter Characteristics
Proc, clutter,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

```

entry_9 = NRLx (Entry, menu,
Label, "Cancellation Characteristics",
proc, canchar,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_19 = NRLx (Entry, menu,
Label, "Title for printout & plot
Proc, title,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_10 = NRLx (Entry, menu,
Label, "Target Velocity Limits
Proc, tarvelo,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_11 = NRLx (Entry, menu,
Label, "PD Contour Plot Data
Proc, contour,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_12 = NRLx (Entry, menu,
Label, "Extended Surface Target Parameters
Proc, surtar,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_13 = NRLx (Entry, menu,
Label, "Parameters for Isometric Plot
Proc, isompt,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_14 = NRLx (Entry, menu,
Label, "Antenna Pattern Print/Plot Parameters
Proc, antpat,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_15 = NRLx (Entry, menu,
Label, "Atmospheric Attenuation Parameters
Proc, atmattn,
Foreground, "black",
NULL);

```

" ,

```

Background, "white",
Shape, oval,
NULL);
```

" ,

```

entry_16 = NRLx (Entry, menu,
Label, "Rain Attenuation Parameters
Proc, rainatt,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);
```

" ,

```

entry_17 = NRLx (Entry, menu,
Label, "Jammer Parameters
Proc, jammer,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);
```

" ,

```

entry_18 = NRLx (Entry, menu,
Label, "Execute - This should be the last one in each case
Proc," execute,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);
```

" ,

```

entry_30 = NRLx (Entry, menu,
Label, "open a new data file
Proc, openfile,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);
```

" ,

```

entry_31 = NRLx (Entry, menu,
Label, "Save this new window
Proc, savewin,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);
```

" ,

```

entry_32 = NRLx (Entry, menu,
Label, "Close the new data file
Proc, closefile,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);
```

" ,

```

entry_40 = NRLx (Entry, menu,
Label, "Open an existing data file
Proc, openinfile,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);
```

```

entry_42 = NRLx (Entry, menu,
Label, "Save this in window",
Proc, saveinf,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_41 = NRLx (Entry, menu,
Label, "close this existing data file",
Proc, closeinfile,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_33 = NRLx (Entry, menu,
Label, "QUIT",
Proc, quit,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

NRLxStartInterface (console);
}

/* Popup Window 2 - Signal Parameters */
void signal()
{
    ycount=5, inc=70, xcount=10;
    i1=i2;i3=14=0;
    v1=v2=v3=v4=v5=v6=0;
    popup2 = NRLx (Popup, console, xPos,530, yPos,20,
Label, "Signal Parameters",
Width, 600,
Height, 800,
Background, "white",
Framecolor, "red",
NULL);
    text200 = NRLx (TextEntry, popup2, xPos, xcount, yPos, ycount,
Label, "Item 1",
Width, 550,
Height, 25,
Proc, sto1,
NULL);
    text = NRLx (TextEntry, popup2, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "RF Transmit Frequency (MHz) ",
Width, 550,
Height, 25,
Proc, sto5,
NULL);
}

void signal()
{
    ycount=5, inc=70, xcount=10;
    i1=i2;i3=14=0;
    v1=v2=v3=v4=v5=v6=0;
    popup2 = NRLx (Popup, console, xPos,530, yPos,20,
Label, "Signal Parameters",
Width, 600,
Height, 800,
Background, "white",
Framecolor, "red",
NULL);
    text200 = NRLx (TextEntry, popup2, xPos, xcount, yPos, ycount,
Label, "Item 1",
Width, 550,
Height, 25,
Proc, sto1,
NULL);
    text = NRLx (TextEntry, popup2, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "RF Transmit Frequency (MHz) ",
Width, 550,
Height, 25,
Proc, sto5,
NULL);
}

void signal()
{
    ycount=5, inc=70, xcount=10;
    i1=i2;i3=14=0;
    v1=v2=v3=v4=v5=v6=0;
    popup2 = NRLx (Popup, console, xPos,530, yPos,20,
Label, "System Noise Temp. (K) ",
Width, 550,
Height, 25,
Proc, sto3,
NULL);
    text201 = NRLx (TextEntry, popup2, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Cumulative System Losses (dB) ",
Width, 550,
Height, 25,
Proc, sto1,
NULL);
    text = NRLx (TextEntry, popup2, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "System Noise Temp. (K) ",
Width, 550,
Height, 25,
Proc, sto3,
NULL);
    text202 = NRLx (TextEntry, popup2, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Instantaneous Bandwidth (MHz) ",
Width, 550,
Height, 25,
Proc, sto1,
NULL);
    text = NRLx (TextEntry, popup2, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Peak Power (MW) ",
Width, 550,
Height, 25,
Proc, sto1,
NULL);
    text203 = NRLx (TextEntry, popup2, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Var 6",
Width, 550,
Height, 25,
Proc, sto6,
NULL);
}

```

```

/* read in an existing file, make changes on card 2 */
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%f%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno !=2) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%f%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
        loc = ftell (fp1);
        printf("win 2 1st loc = %d\n", loc);
        loc = loc - 80;
        printf("win 2 2nd loc=%d\n", loc);
        sprintf (s1, "%d", 11);
        sprintf (s2, "%d", 12);
        sprintf (s3, "%d", 13);
        sprintf (s4, "%d", 14);
        sprintf (sv1, "%f", v1);
        sprintf (sv2, "%f", v2);
        sprintf (sv3, "%f", v3);
        sprintf (sv4, "%f", v4);
        sprintf (sv5, "%f", v5);
        sprintf (sv6, "%f", v6);
        NRLXLoadTextEntry (text200, s1);
        NRLXLoadTextEntry (text, s12);
        NRLXLoadTextEntry (text201, s13);
        NRLXLoadTextEntry (text202, s14);
        NRLXLoadTextEntry (text1, sv1);
        NRLXLoadTextEntry (text2, sv2);
        NRLXLoadTextEntry (text3, sv3);
        NRLXLoadTextEntry (text4, sv4);
        NRLXLoadTextEntry (text5, sv5);
        NRLXLoadTextEntry (text203, sv6);
    }
}

/* Popup Window 3 - Radar Detection Parameters */
void radardet()
{
    ycount=5, inc=70, xcount=10;
    cardno=3, i1=i2-i3-i4=0;
    v1=v2=v3=v4=v5=v6=0;
    popup3 = NRLX (popup, console, xPos, 530, yPos, 10,
Label, "Radar Detection Parameters",
Width, 600,
Height, 800,

```

```

Label, "Detection Sensitivity (dB)",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text14 = NRLX (TextEntry, Popup4, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Detection Slope (default=1)",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

/* read in an existing file, make changes on card 3 */
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "#d#d#d#d#f#f#f#f#f", &cardno, &il, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno != 3) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "#d#d#d#d#f#f#f#f#f", &cardno, &il, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }
}

loc = ftell (fp1);
printf("win 3 1st loc = %d\n", loc);
printf("win 3 2nd loc=%d\n", loc);

sprintf ("sil", "%d", i1);
sprintf ("i2", "%d", i2);
sprintf ("i3", "%d", i3);
sprintf ("i4", "%d", i4);
sprintf ("sv1", "%f", v1);
sprintf ("sv2", "%f", v2);
sprintf ("sv3", "%f", v3);
sprintf ("sv4", "%f", v4);
sprintf ("sv5", "%f", v5);
sprintf ("sv6", "%f", v6);

NRLXLoadTextEntry (text6, si1);
NRLXLoadTextEntry (text7, si2);
NRLXLoadTextEntry (text8, si3);
NRLXLoadTextEntry (text9, si4);
NRLXLoadTextEntry (text20, sv1);
NRLXLoadTextEntry (text10, sv2);
NRLXLoadTextEntry (text11, sv3);
NRLXLoadTextEntry (text12, sv4);
NRLXLoadTextEntry (text13, sv5);
NRLXLoadTextEntry (text14, sv6);

/* Popup Window 4 - AEW Geometry */
void aewgeo()
{
    ycount=5, inc=70, xcount=10;
    cardno=4;
    il=i2;i3=14=0;
    v1=v2-v3=v4=v5=v6=0;
    popup4 = NRLX (Popup, console, XPos,530, YPos, 20,
Label, "AEW Geometry",
Height, 800,
Width, 600,
Background, "white",
FrameColor, "green",
NULL);

    text205 = NRLX (TextEntry, Popup4, XPos, xcount, YPos, ycount,
Label, "Item 1",
Width, 550,
Height, 25,
Proc, stoi1,
NULL);

    text206 = NRLX (TextEntry, Popup4, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Item 2",
Width, 550,
Height, 25,
Proc, stoi2,
NULL);

    text207 = NRLX (TextEntry, Popup4, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
NULL);

    text208 = NRLX (TextEntry, Popup4, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, stoi3,
NULL);

    text15 = NRLX (TextEntry, Popup4, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Platform Altitude (ft.)",
Width, 550,
Height, 25,
Proc, stoi1,
NULL);

    text16 = NRLX (TextEntry, Popup4, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Platform Pitch Angle (deg.)",
Width, 550,
Height, 25,
Proc, stoi2,
NULL);

    text17 = NRLX (TextEntry, Popup4, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Platform Roll Angle (deg.)",
Width, 550,
Height, 25,
NULL);
}

```

```

PROC, stof3,
NULL);

text209 = NRLX (TextEntry, popup4, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 4",
Width, 550,
Height, 25,
proc, stof4,
NULL);

text210 = NRLX (TextEntry, popup4, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 5",
Width, 550,
Height, 25,
proc, stof5,
NULL);

text211 = NRLX (TextEntry, popup4, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 6",
Width, 550,
Height, 25,
proc, stof6,
NULL);

/* read in an existing file, make changes on card 4 */

if (cndt == 1) {
    rewind (fpl);
    fscanf (fpl, "#d#d#d#d#f#f#f#f", &cardno, s11, s12, s13, s14, sv
1, sv2, sv3, sv4, sv5, sv6);
    while (cardno != 1) {
        fseek (fpl, 1, 1);
        fscanf (fpl, "#d#d#d#d#f#f#f#f", &cardno, s11, s12, s13, s14, sv
1, sv2, sv3, sv4, sv5, sv6);
    }
    loc = ftell (fpl);
    printf ("Win 4 1st loc = %d\n", loc);
    loc = loc - 80;
    printf ("Win 4 2nd loc=%d\n", loc);
    sprintf (s11, "%d", i1);
    sprintf (s12, "%d", i2);
    sprintf (s13, "%d", i3);
    sprintf (s14, "%d", i4);
    sprintf (sv1, "%f", v1);
    sprintf (sv2, "%f", v2);
    sprintf (sv3, "%f", v3);
    sprintf (sv4, "%f", v4);
    sprintf (sv5, "%f", v5);
    sprintf (sv6, "%f", v6);
    NRLXLoadTextEntry (text205, s11);
    NRLXLoadTextEntry (text206, s12);
    NRLXLoadTextEntry (text207, s13);
    NRLXLoadTextEntry (text208, s14);
    NRLXLoadTextEntry (text15, sv1);
    NRLXLoadTextEntry (text16, sv2);
    /* Popup Window 5 - Target Height Limits */
    void targt()
    {
        ycount=5, inc=70, xcount=10;
        cardno=5;
        i1=i2=i3=i4=0;
        v1=v2=v3=v4=v5=v6=0;
        popup5 = NRLX (Popup, console, xpos 530, ypos, 20,
Label, "Target Height Limits",
width, 600,
height, 200,
background, "white",
framecolor, "gold",
NULL);
        text212 = NRLX (TextEntry, popup5, xpos, xcount, ypos, ycount,
Label, "Item 1",
width, 550,
height, 25,
proc, sto1,
NULL);
        text213 = NRLX (TextEntry, popup5, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Item 2",
width, 550,
height, 25,
proc, sto2,
NULL);
        text214 = NRLX (TextEntry, popup5, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Item 3",
width, 550,
height, 25,
proc, sto3,
NULL);
        text215 = NRLX (TextEntry, popup5, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Item 4",
width, 550,
height, 25,
proc, sto4,
NULL);
        text18 = NRLX (TextEntry, popup5, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Minimum Target Altitude (ft.)",
width, 550,
height, 25,
proc, sto1,
NULL);
    }
}

```

```

text19 = NRLX (TextEntry, popup5, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Maximum Target Altitude (ft.)",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text20 = NRLX (TextEntry, popup5, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Increment Altitude (ft.)",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text216 = NRLX (TextEntry, popup5, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text217 = NRLX (TextEntry, popup5, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 5",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text218 = NRLX (TextEntry, popup5, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 6",
Width, 550,
Height, 25,
NULL);

/* read in an existing file, make changes on card 5 */

if (cndt == 1) {
    if (fread (fp1, 1, 1, &v1, &v2, &v3, &v4, &v5, &v6);
        rewind (fp1);
        while (cardno <= 5) {
            fscanf (fp1, "%d%d%d%d%d%d", &cardno, &v1, &v2, &v3, &v4, &v5, &v6);
            loc = ftell (fp1);
            printf ("%win 5 1st loc = %d\n", loc);
            loc = loc + 80;
            printf ("%win 5 2nd loc=%d\n", loc);
            }
        }

loc = ftell (fp1);
printf ("%win 5 1st loc = %d\n", loc);
loc = loc + 80;
printf ("%win 5 2nd loc=%d\n", loc);

sprint (sv1, "%d", i1);
sprint (sv2, "%d", i2);
sprint (sv3, "%d", i3);
sprint (sv4, "%d", i4);
sprint (sv1, "%f", v1);
sprint (sv2, "%f", v2);
sprint (sv3, "%f", v3);
sprint (sv4, "%f", v4);
sprint (sv5, "%f", v5);
sprint (sv6, "%f", v6);

NRLXLoadNextEntry (text212, si1);
NRLXLoadNextEntry (text213, si2);
NRLXLoadNextEntry (text214, si3);
NRLXLoadNextEntry (text215, si4);
NRLXLoadNextEntry (text218, sv1);
NRLXLoadNextEntry (text19, sv2);
NRLXLoadNextEntry (text20, sv3);
NRLXLoadNextEntry (text216, sv4);
NRLXLoadNextEntry (text217, sv5);
NRLXLoadNextEntry (text218, sv6);
}
}

/* Popup Window 6 - Target Range Limits */
void tarrng()
{
ycount=5, inc=70, xcount=10;
cardno=6;
i1=i2;i3=i4=0;
v1=v2=v3=v4=v5=v6=0;
popup6 = NRLX (POPUP, console, Xpos, 530, Ypos, 20,
Label, "Target Range Limits",
Width, 600,
Height, 800,
Background, "white",
FrameColor, "magenta",
NULL);
text219 = NRLX (TextEntry, popup6, xpos, xcount, ypos, ycount,
Label, "Item 1",
Width, 550,
Height, 25,
Proc, stoi1,
NULL);
text220 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Item 2",
Width, 550,
Height, 25,
Proc, stoi2,
NULL);
text221 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
Proc, stoi3,
NULL);
text222 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, stoi4,
NULL);
}

```

```

Width, 550,
Height, 25,
Proc, stoi4,
NULL);

text21 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount-ycount+inc),
Label, "Minimum Target Range (nmi)",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

text22 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount-ycount+inc),
Label, "Maximum Target Range (nmi)",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text23 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount-ycount+inc),
Label, "Increment Range (nmi)",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text223 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount-ycount+inc),
Label, "Var 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text224 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount-ycount+inc),
Label, "Var 5",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text225 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount-ycount+inc),
Label, "Var 6",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 6 */
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%ff%ff%ff", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
}

while (cardno != 6) {
    fseek (fp1, 1, 1);
    fscanf (fp1, "%d%d%d%d%ff%ff%ff", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
}

```

```

loc = ftell (fp1);
printf ("win 6 1st loc = %d\n", loc);
loc = 80;
printf ("win 6 2nd loc=%d\n", loc);
sprintf (s1, "%d", 11);
sprintf (s2, "%d", 12);
sprintf (s3, "%d", 13);
sprintf (s4, "%d", 14);
sprintf (sv1, "%f", v1);
sprintf (sv2, "%f", v2);
sprintf (sv3, "%f", v3);
sprintf (sv4, "%f", v4);
sprintf (sv5, "%f", v5);
sprintf (sv6, "%f", v6);

NRLXLoadTextEntry (text219, s11);
NRLXLoadTextEntry (text220, s12);
NRLXLoadTextEntry (text221, s13);
NRLXLoadTextEntry (text222, s14);
NRLXLoadTextEntry (text223, sv1);
NRLXLoadTextEntry (text224, sv2);
NRLXLoadTextEntry (text225, sv3);
NRLXLoadTextEntry (text226, sv4);
NRLXLoadTextEntry (text227, sv5);
NRLXLoadTextEntry (text228, sv6);

/* Popup Window 7 - Antenna Characteristics */
void antcha()
{
    ycount=5, inc=70, xcount=10;
    cardno=7;
    i1=i2=13=i4=0;
    v1=v2=v3=v4=v5=v6=0;
    popup7 = NRLX (Popup, console, xpos, 530, ypos, 10,
Label, "Antenna Characteristics",
Width, 600,
Height, 800,
Background, "white",
FrameColor, "dark turquoise",
NULL);

text24 = NRLX (TextEntry, popup7, xPos, xcount, yPos, yCount,
Label, "# of Antenna Pattern Data Points",
Width, 550,
Height, 25,
Proc, stoi6,
NULL);

text25 = NRLX (TextEntry, popup7, xPos, xcount, yPos, yCount+inc),
Label, "Reference Count (10*full scale in dB)",
Width, 550,
Height, 25,

```

```

Proc, stoi2,
NULL);

text26 = NRLX (TextEntry, popup7, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "0 - Horizontal Polarity, 1 - Vertical Polarity",
Width, 550,
Height, 25,
Proc, stoi3,
NULL);

text27 = NRLX (TextEntry, popup7, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "0-T/R Parameter,1-Transmit pattern,2-Receive pattern,3-vPA
T routine",
Width, 550,
Height, 25,
Proc, stoi4,
NULL);

text28 = NRLX (TextEntry, popup7, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Antenna az. pointing angle (deg.)",
Width, 550,
Height, 25,
Proc, stoi1,
NULL);

text29 = NRLX (TextEntry, popup7, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Antenna Reference Gain (dB)",
Width, 550,
Height, 25,
Proc, stoi2,
NULL);

text30 = NRLX (TextEntry, popup7, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Min Angle of Antenna Pattern(if I1!=0), Vertical Beamwidth
h (deg)",
Width, 550,
Height, 25,
Proc, stori3,
NULL);

text31 = NRLX (TextEntry, popup7, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Increment of Pattern (if I1 != 0) ,
Width, 550,
Height, 25,
Proc, stoi4,
NULL);

text32 = NRLX (TextEntry, popup7, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Full scale of Antenna Chart (dB)(if I1!=0), Elevation off
set Angle (deg)",
Width, 550,
Height, 25,
Proc, stori5,
NULL);

text33 = NRLX (TextEntry, popup7, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Vertical Pattern Phase Parameter",
Width, 550,
Height, 25,
Proc, stoi6,
NULL);
/* read in an existing file, make changes on card 7 */
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &y
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno !=7) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &y
1, &v2, &v3, &v4, &v5, &v6);
    }
    loc = ftell (fp1);
    printf("win 7 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 7 2nd loc=%d\n", loc);
    sprintf ("s1, \"d\", i1);
    sprintf ("s12, \"d\", i2);
    sprintf ("s13, \"d\", i3);
    sprintf ("s14, \"d\", i4);
    sprintf ("sv1, \"f\", v1);
    sprintf ("sv2, \"f\", v2);
    sprintf ("sv3, \"f\", v3);
    sprintf ("sv4, \"f\", v4);
    sprintf ("sv5, \"f\", v5);
    sprintf ("sv6, \"f\", v6);
NRLXLoadTextEntry (text24, s1);
NRLXLoadTextEntry (text25, s12);
NRLXLoadTextEntry (text26, s13);
NRLXLoadTextEntry (text27, s14);
NRLXLoadTextEntry (text28, sv1);
NRLXLoadTextEntry (text29, sv2);
NRLXLoadTextEntry (text30, sv3);
NRLXLoadTextEntry (text31, sv4);
NRLXLoadTextEntry (text32, sv5);
NRLXLoadTextEntry (text33, sv6);
}
/* Popup Window 8 - Stored Antenna Data Points */
void antdata()
{
    ycount=5, inc=70, xcount=10;
    cardno=8;
    i1=12=i3;i4=0;
    v1=v2=v3=v4=v5=v6=0;
    popup8 = NRLX (Popup, console, XPOS, 530, YPOS, 20,
Label, "Stored Antenna Data Points",
Width, 600,
Height, 800,
Background, "white",

```

```

FrameColor, "violet red",
NULL);

text34 = NRLX (TextEntry, popup8, xPos, xcount, yPos, ycount,
Label, "# of Current Antenna Point",
Width, 550,
Height, 25,
Proc, sto1,
NULL);

text35 = NRLX (TextEntry, popup8, xPos, xcount, yPos, (ycount-ycount+inc),
Label, "Recorder Count (10^value in dB)",
Width, 550,
Height, 25,
Proc, sto2,
NULL);

text226 = NRLX (TextEntry, popup8, xPos, xcount, yPos, (ycount-ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
Proc, sto13,
NULL);

text227 = NRLX (TextEntry, popup8, xPos, xcount, yPos, (ycount-ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, sto14,
NULL);

text36 = NRLX (TextEntry, popup8, xPos, xcount, yPos, (ycount-ycount+inc),
Label, "Voltage Pattern Sign (0, +1, -1)",
Width, 550,
Height, 25,
Proc, sto1,
NULL);

text228 = NRLX (TextEntry, popup8, xPos, xcount, yPos, (ycount-ycount+inc),
Label, "Var 2",
Width, 550,
Height, 25,
Proc, sto2,
NULL);

text229 = NRLX (TextEntry, popup8, xPos, xcount, yPos, (ycount-ycount+inc),
Label, "Var 3",
Width, 550,
Height, 25,
Proc, sto3,
NULL);

text230 = NRLX (TextEntry, popup8, xPos, xcount, yPos, (ycount-ycount+inc),
Label, "Var 4",
Width, 550,
Height, 25,
Proc, sto4,
NULL);

text231 = NRLX (TextEntry, popup8, xPos, xcount, yPos, (ycount-ycount+inc),
Label, "Var 5",
Width, 550,
Height, 25,
Proc, sto5,
NULL);

```

B-13

```

width, 550,
height, 25,
proc, sto5,
NULL);

text232 = NRLX (TextEntry, popup8, xPos, xcount, yPos, (ycount-ycount+inc),
Label, "var 6",
Width, 550,
Height, 25,
Proc, sto6,
NULL);

/* read in an existing file, make changes */
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno != 8) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d", &i1);
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d", &i2);
        loc = ftell (fp1);
        printf ("win 8 1st loc = %d\n", loc);
        loc = loc - 80;
        print ("win 8 2nd loc=%d\n", loc);
        sprintf ("%d", i1);
        sprintf ("%d", i2);
        sprintf ("%d", i3);
        sprintf ("%d", i4);
        sprintf ("%d", i5);
        sprintf ("%d", i6);
        sprintf ("%f", v1);
        sprintf ("%f", v2);
        sprintf ("%f", v3);
        sprintf ("%f", v4);
        sprintf ("%f", v5);
        sprintf ("%f", v6);
        NRLXloadTextEntry (text34, s11);
        NRLXloadTextEntry (text35, s12);
        NRLXloadTextEntry (text36, s13);
        NRLXloadTextEntry (text227, s14);
        NRLXloadTextEntry (text228, s15);
        NRLXloadTextEntry (text229, s16);
        NRLXloadTextEntry (text230, s17);
        NRLXloadTextEntry (text231, s18);
        NRLXloadTextEntry (text232, s19);
    }
}

/* Popup Window 9 - Clutter Characteristics */
void clitchar()
{
}

```

```

ycount=5, inc=70, xcount=3;
cardno=9;
i1=i2=3=i4=0;
v1=v2=v3=v4=v5=v6=0;

popUp9 = NRLX (Popup, console, xPos, yPos, 20,
Label, "Clutter Characteristics",
Width, 600,
Height, 800,
Background, "White",
FrameColor, "blue",
NULL);

text37 = NRLX (TextEntry, popup9, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "0-Land,1-Sea,Model Clutter Coeff func:2/3(L/S)-range, 4/5
(L/S)-grazing angle",
Width, 580,
Height, 25,
Proc, sto1,
NULL);

text38 = NRLX (TextEntry, popup9, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Diffraction option: 0-Interpolate,1-Multi Path,2-Fock",
Width, 580,
Height, 25,
NULL);

text39 = NRLX (TextEntry, popup9, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "0-(all cases except 2/3),Index# of Range Interval for coe
ff",
Width, 580,
Height, 25,
Proc, sto3,
NULL);

text40 = NRLX (TextEntry, popup9, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "0-No Forward Scatter,1-Forward Reflection,Forward Power:2
-Min,3-Max",
Width, 580,
Height, 25,
Proc, sto4,
NULL);

text41 = NRLX (TextEntry, popup9, xPos, xcount, yPos, (ycount=ycount+inc),
at 10 (deg),
Width, 580,
Height, 25,
Proc, sto5,
NULL);

text42 = NRLX (TextEntry, popup9, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Horizontal Beamwider (deg.)",
Width, 580,
Height, 25,
Proc, sto6,
NULL);

text43 = NRLX (TextEntry, popup9, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Pulse width (past-compressed) (micro sec.),
```

```

NRLxLoadTextEntry (text37, si1);
NRLxLoadTextEntry (text38, si2);
NRLxLoadTextEntry (text39, si3);
NRLxLoadTextEntry (text40, si4);
NRLxLoadTextEntry (text41, sv1);
NRLxLoadTextEntry (text42, sv2);
NRLxLoadTextEntry (text43, sv3);
NRLxLoadTextEntry (text44, sv4);
NRLxLoadTextEntry (text45, sv5);
NRLxLoadTextEntry (text46, sv6);

/*
 Popup Window 10 - Cancellation Characteristics */
void canchar()
{
    ycount=5, inc=70, xcount=3;
    cardno=10;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup10 = NRLx (Popup, console, xpos,530, ypos, 20,
                     Label, "Cancellation Characteristics",
                     width, 600,
                     height, 800,
                     background, "white",
                     framecolor, "medium sea green",
                     NULL);

    text47 = NRLx (TextEntry, popup10, xpos, xcount, ypos, ycount,
                    Label, "0-single Cancellation ratio, i-Cancellation ratio for #th
range value",
                    width, 580,
                    height, 25,
                    proc, sto1,
                    NULL);

    text48 = NRLx (TextEntry, popup10, xpos, xcount, ypos, ycount="ycount+inc",
                    Label, "# - Pulses per Antenna Beamwidth",
                    width, 580,
                    height, 25,
                    proc, sto2,
                    NULL);

    text49 = NRLx (TextEntry, popup10, xpos, xcount, ypos, ycount="ycount+inc",
                    Label, "-0-No Loss func.,1-Single Delay Canceller,2-double Delay C /* read in an existing file, make changes on card 10*/
anceller",
                    width, 580,
                    height, 25,
                    proc, sto3,
                    NULL);

    text233 = NRLx (TextEntry, popup10, xpos, xcount, ypos, ycount="ycount+inc",
                    Label, "Item 4",
                    width, 580,
                    height, 25,
                    proc, sto4,
                    NULL);
}

text50 = NRLx (TextEntry, popup10, xpos, xcount, ypos, ycount="ycount+inc",
               Label, "Cancellation Ratio (dB)",
               width, 580,
               height, 25,
               proc, sto1,
               NULL);

text51 = NRLx (TextEntry, popup10, xpos, xcount, ypos, ycount="ycount+inc",
               Label, "Threshold Loss Coefficient - A1 (if unknown use 0)",
               width, 580,
               height, 25,
               proc, sto2,
               NULL);

text52 = NRLx (TextEntry, popup10, xpos, xcount, ypos, ycount="ycount+inc",
               Label, "Threshold Loss Coefficient - A2 (if unknown use 5)",
               width, 580,
               height, 25,
               proc, sto3,
               NULL);

text53 = NRLx (TextEntry, popup10, xpos, xcount, ypos, ycount="ycount+inc",
               Label, "Threshold Loss Coefficient - B1 (if unknown use 1)",
               width, 580,
               height, 25,
               proc, sto4,
               NULL);

text54 = NRLx (TextEntry, popup10, xpos, xcount, ypos, ycount="ycount+inc",
               Label, "Threshold Loss Coefficient - B2",
               width, 580,
               height, 25,
               proc, sto5,
               NULL);

text55 = NRLx (TextEntry, popup10, xpos, xcount, ypos, ycount="ycount+inc",
               Label, "Threshold Loss Coefficient - B3",
               width, 580,
               height, 25,
               proc, sto6,
               NULL);

while (cardno !=10) {
    fseek (fp1, 1, 1);
    fscanf (fp1, "%d%d%d%d%ff%ff%ff", &cardno, &i1, &i2, &i3, &i4, &v1,
            &v2, &v3, &v4, &v5, &v6);
}

```

```

loc = ftell(fp1);
printf("win 10 1st loc = %d\n", loc);
loc = loc - 80;
printf("win 10 2nd loc=%d\n", loc);

sprintf(sil, "%d", i1);
sprintf(s12, "%d", i2);
sprintf(s13, "%d", i3);
sprintf(s14, "%d", i4);
sprintf(sv1, "%f", v1);
sprintf(sv2, "%f", v2);
sprintf(sv3, "%f", v3);
sprintf(sv4, "%f", v4);
sprintf(sv5, "%f", v5);
sprintf(sv6, "%f", v6);

NRLxLoadTextEntry(text47, s11);
NRLxLoadTextEntry(text48, s12);
NRLxLoadTextEntry(text49, s13);
NRLxLoadTextEntry(text233, s14);
NRLxLoadTextEntry(text50, sv1);
NRLxLoadTextEntry(text51, sv2);
NRLxLoadTextEntry(text52, sv3);
NRLxLoadTextEntry(text53, sv4);
NRLxLoadTextEntry(text54, sv5);
NRLxLoadTextEntry(text55, sv6);
}

/* Popup Window 12 - Target Velocity Limits */
void tarvelo()
{
    ycount=5, inc=70, xcount=10;
    carno=12;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup12 = NRLx (popup, console, xPos, 530, yPos, 20,
                    Label, "Target Velocity Limits",
                    Width, 600,
                    Height, 800,
                    Background, "White",
                    FrameColor, "Dark Orchid",
                    NULL);

    text56 = NRLx (TextEntry, popup12, xPos, xcount, yPos, ycount,
                    Label, "0-For Regular PD Calc, 1-For Cumulative PD Calc",
                    Width, 580,
                    Height, 25,
                    Proc, sto1,
                    NULL);

    text214 = NRLx (TextEntry, popup12, xPos, xcount, yPos, (ycount=ycount+inc),
                    Label, "Item 2",
                    Width, 580,
                    Height, 25,
                    Proc, sto12,
                    NULL);
}

ycount=5, inc=70, xcount=10;
carno=12;
i1=i2=i3=i4=0;
v1=v2=v3=v4=v5=v6=0;

popup12 = NRLx (popup, console, xPos, 530, yPos, 20,
Label, "Target Velocity Limits",
Width, 600,
Height, 800,
Background, "White",
FrameColor, "Dark Orchid",
NULL);

text56 = NRLx (TextEntry, popup12, xPos, xcount, yPos, ycount,
Label, "0-For Regular PD Calc, 1-For Cumulative PD Calc",
Width, 580,
Height, 25,
Proc, sto1,
NULL);

text215 = NRLx (TextEntry, popup12, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Item 3",
Width, 580,
Height, 25,
Proc, sto13,
NULL);

text236 = NRLx (TextEntry, popup12, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Item 4",
Width, 580,
Height, 25,
Proc, sto14,
NULL);

text57 = NRLx (TextEntry, popup12, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Minimum Target Range Rate (ft/sec)",
Width, 580,
Height, 25,
Proc, sto15,
NULL);

text58 = NRLx (TextEntry, popup12, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Maximum Target Range Rate (ft/sec)",
Width, 580,
Height, 25,
Proc, sto16,
NULL);

text59 = NRLx (TextEntry, popup12, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Increment Rate (ft/sec)",
Width, 580,
Height, 25,
Proc, sto17,
NULL);

text60 = NRLx (TextEntry, popup12, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Scan Interval (sec)",
Width, 580,
Height, 25,
Proc, sto18,
NULL);

text61 = NRLx (TextEntry, popup12, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Range Increment Between Printouts (rmi)",
Width, 580,
Height, 25,
Proc, sto19,
NULL);

text62 = NRLx (TextEntry, popup12, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Cumulative Prob Level for Special Print",
Width, 580,
Height, 25,
Proc, sto20,
NULL);

/* read in an existing file, make changes on card 12 */
if (cnrdt == 1) {
}

```

Wed Sep 21 16:37:05 1994

```

rewind (fp1);
fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v1,
        &v2, &v3, &v4, &v5, &v6);
while (cardno !=12) {
    fseek (fp1, 1, 1);
    fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v1,
            &v2, &v3, &v4, &v5, &v6);

    loc = ftell (fp1);
    printf("win 12 1st loc = %d\n", loc);
    print("win 12 2nd loc=%d\n", loc);

    sprintf (s11, "%d", i1);
    sprintf (s12, "%d", i2);
    sprintf (s13, "%d", i3);
    sprintf (s14, "%d", i4);
    sprintf (sv1, "%f", v1);
    sprintf (sv2, "%f", v2);
    sprintf (sv3, "%f", v3);
    sprintf (sv4, "%f", v4);
    sprintf (sv5, "%f", v5);
    sprintf (sv6, "%f", v6);

    NRLXLoadTextEntry (text56, s11);
    NRLXLoadTextEntry (text23, s12);
    NRLXLoadTextEntry (text235, s13);
    NRLXLoadTextEntry (text236, s14);
    NRLXLoadTextEntry (text57, sv1);
    NRLXLoadTextEntry (text58, sv2);
    NRLXLoadTextEntry (text59, sv3);
    NRLXLoadTextEntry (text60, sv4);
    NRLXLoadTextEntry (text61, sv5);
    NRLXLoadTextEntry (text62, sv6);
}

/*
 * Popup Window 13 - PD Contour Plot Data */
void contour()
{
    ycount=5, inc=70, xcount=10;
    cardno=13;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;
    popup13 = NRLX (Popup, console, xpos, 510, ypos, 20,
                    Label, "PD Contour Plot Data",
                    width, 600,
                    height, 800,
                    background, "White",
                    framecolor, "brown",
                    NULL);

    text63 = NRLX (TextEntry, popup13, xpos, xcount, ypos, ycount,
                    label, "0-Regular Plots only, 1-Automatic Contour Plots",
                    width, 580,
                    height, 25,
                    proc, sto1,
                    NULL);

    text64 = NRLX (TextEntry, popup13, xpos, xcount, ypos, (ycount-ycount+inc),
                    label, "1 of Contour Levels (1+1/step_size)",
                    width, 580,
                    height, 25,
                    proc, sto2,
                    NULL);

    text237 = NRLX (TextEntry, popup13, xpos, xcount, ypos, (ycount-ycount+inc),
                    label, "Item 3",
                    width, 580,
                    height, 25,
                    proc, sto3,
                    NULL);

    text238 = NRLX (TextEntry, popup13, xpos, xcount, ypos, (ycount-ycount+inc),
                    label, "Item 4",
                    width, 580,
                    height, 25,
                    proc, sto4,
                    NULL);

    text65 = NRLX (TextEntry, popup13, xpos, xcount, ypos, (ycount-ycount+inc),
                    label, "Maximum Target Altitude (ft.)",
                    width, 580,
                    height, 25,
                    proc, sto5,
                    NULL);

    text66 = NRLX (TextEntry, popup13, xpos, xcount, ypos, (ycount-ycount+inc),
                    label, "Range Scale (mi/in.)",
                    width, 580,
                    height, 25,
                    proc, sto2,
                    NULL);

    text67 = NRLX (TextEntry, popup13, xpos, xcount, ypos, (ycount-ycount+inc),
                    label, "Height Scale (ft/in.)",
                    width, 580,
                    height, 25,
                    proc, sto3,
                    NULL);

    text239 = NRLX (TextEntry, popup13, xpos, xcount, ypos, (ycount-ycount+inc),
                    label, "Var 4",
                    width, 580,
                    height, 25,
                    proc, sto4,
                    NULL);

    text240 = NRLX (TextEntry, popup13, xpos, xcount, ypos, (ycount-ycount+inc),
                    label, "Var 5",
                    width, 580,
                    height, 25,
                    proc, sto5,
                    NULL);
}

```

```

    Proc, stoff,
    NULL);

text241 = NRLx ("TextEntry, popup13, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 6",
Width, 580,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 13*/
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "#d#d#d#d#f#f#f#f#f#f", &cardno, &il1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno !=13) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "#d#d#d#d#f#f#f#f#f", &cardno, &il1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }
}

loc = ftell (fp1);
print ("win 13 1st loc = $d\n", loc);
loc = loc - 80;
print ("win 13 2nd loc=$d\n", loc);

sprintf (&il1, "%d", i1);
sprintf (&i2, "%d", i2);
sprintf (&i3, "%d", i3);
sprintf (&i4, "%d", i4);
sprintf (&v1, "%f", v1);
sprintf (&v2, "%f", v2);
sprintf (&v3, "%f", v3);
sprintf (&v4, "%f", v4);
sprintf (&v5, "%f", v5);
sprintf (&v6, "%f", v6);

NRLxLoadTextEntry (text63, si1);
NRLxLoadTextEntry (text64, si2);
NRLxLoadTextEntry (text237, si3);
NRLxLoadTextEntry (text238, si4);
NRLxLoadTextEntry (text65, sv1);
NRLxLoadTextEntry (text66, sv2);
NRLxLoadTextEntry (text67, sv3);
NRLxLoadTextEntry (text239, sv4);
NRLxLoadTextEntry (text240, sv5);
NRLxLoadTextEntry (text241, sv6);
}

/* Popup Window 15 - Extended Surface Target Parameters */
void surtar()
{
ycount=5, inc=70, xcount=10;
cardno=15;
}

```

```

        Label, "Var 4",
        Width, 580,
        Height, 25,
        Proc, stof4,
        NULL);

text245 = NRLx (TextEntry, popup15, xPos, xcount, yPos, (ycount=ycount+inc),
        Label, "Var 5",
        Width, 580,
        Height, 25,
        Proc, stof5,
        NULL);

text246 = NRLx (TextEntry, popup15, xPos, xcount, yPos, (ycount=ycount+inc),
        Label, "Var 6",
        Width, 580,
        Height, 25,
        Proc, stof6,
        NULL);

/* read in an existing file, make changes on card 15 */

if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);

    while (cardno !=15) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }

    loc = ftell (fp1);
    print("win 15 1st loc = %d\n", loc);
    loc = loc - 80;
    print("win 15 2nd loc=%d\n", loc);

    sprint ("s1", "%d", i1);
    sprint ("s12", "%d", i12);
    sprint ("s13", "%d", i13);
    sprint ("s14", "%d", i14);
    sprint ("sv1", "%f", v1);
    sprint ("sv2", "%f", v2);
    sprint ("sv3", "%f", v3);
    sprint ("sv4", "%f", v4);
    sprint ("sv5", "%f", v5);
    sprint ("sv6", "%f", v6);

    NRLxLoadTextEntry (text68, s1);
    NRLxLoadTextEntry (text69, s12);
    NRLxLoadTextEntry (text70, s13);
    NRLxLoadTextEntry (text71, s14);
    NRLxLoadTextEntry (text72, sv1);
    NRLxLoadTextEntry (text73, sv2);
    NRLxLoadTextEntry (text74, sv3);
    NRLxLoadTextEntry (text75, sv4);
    NRLxLoadTextEntry (text76, sv5);
    NRLxLoadTextEntry (text77, sv6);
}

/* Popup Window 16 - Parameters for Isometric Plot */

void isompt()
{
    ycount=5, inc=65, xcount=5, yycount=290;
    cardno=16;
    i1=i2-i3-14=0;
    v1=v2=v3=v4=v5=v6=0;

    popup16 = NRLx (Popup, console, XPos, 530, YPos, 20,
        Label, "Parameters for Isometric Plot",
        Width, 600,
        Height, 800,
        Background, "White",
        FrameColor, "magenta",
        NULL);

    text73 = NRLx (Button, popup16, XPos, xcount, yPos, ycount,
        Label, "Item 1: 0 - None (1,2,3)",
        Proc, item16,
        Data, 0,
        Foreground, "salmon",
        NULL);

    text74 = NRLx (Button, popup16, XPos, 400, YPos, ycount,
        Label, "Item 1 - Signal/Noise",
        Proc, item16,
        Data, 1,
        Foreground, "black",
        NULL);

    text75 = NRLx (Button, popup16, XPos, xcount, yPos, 30,
        Label, "2 - Clutter/Noise",
        Proc, item16,
        Data, 2,
        Foreground, "black",
        NULL);

    text76 = NRLx (Button, popup16, XPos, 300, YPos, 30,
        Label, "3 - Probability of Detection",
        Proc, item16,
        Data, 3,
        Foreground, "black",
        NULL);

    text77 = NRLx (Button, popup16, XPos, ycount, yPos, 55,
        Label, "Height Independent Plots : 10 - Reset to zero(11,12,13)",
        Proc, item16,
        Data, 10,
        Foreground, "black",
        NULL);

    text78 = NRLx (Button, popup16, XPos, xcount, yPos, 80,
        Label, "11 - sh 1",
        Proc, item16,
        Data, 11,
        NULL);
}

```

```

Foreground, "black",
NULL);
text79 = NRLX (Button, popup16, xPos, 200, yPos, 80,
Label, "12 - sh 2",
Proc, item6,
Data, 12,
Foreground, "black",
NULL);

text80 = NRLX (Button, popup16, xPos, 400, yPos, 80,
Label, "13 - sh 3",
Proc, item6,
Data, 13,
Foreground, "black",
NULL);

text81 = NRLX (Button, popup16, xPos, xcount, yPos, 105,
Label, "Height Dependent Plots: 20 - Reset to zero(21,22,23)",
Proc, item6,
Data, 20,
Foreground, "black",
NULL);

text82 = NRLX (Button, popup16, xPos, xcount, yPos, 130,
Label, "21 - sh1",
Proc, item6,
Data, 21,
Foreground, "black",
NULL);

text83 = NRLX (Button, popup16, xPos, 200, yPos, 130,
Label, "22 - sh2",
Proc, item6,
Data, 22,
Foreground, "black",
NULL);

text84 = NRLX (Button, popup16, xPos, 400, yPos, 130,
Label, "23 - sh3",
Proc, item6,
Data, 23,
Foreground, "black",
NULL);

text85 = NRLX (Button, popup16, xPos, xcount, yPos, 160,
Label, "Item 2: 0 - Reset",
Proc, item2,
Data, 0,
Foreground, "salmon",
NULL);

text86 = NRLX (Button, popup16, xPos, 200, yPos, 160,
Label, "Height Independent Plots: 1 - Clutter/Noise",
Proc, item2,
Data, 1,
Foreground, "black",
NULL);

text87 = NRLX (Button, popup16, xPos, xcount, yPos, 185,
Label, "2 - Threshold Loss",
Proc, item2,
Foreground, "black",
NULL);

Data, 2,
Foreground, "black",
NULL);

text88 = NRLX (Button, popup16, xPos, 400, yPos, 185,
Label, "3 - Cancellation Ratio",
Proc, item2,
Data, 3,
Foreground, "black",
NULL);

text89 = NRLX (Button, popup16, xPos, xcount, yPos, 215,
Label, "Height Dependent Plots: 1 - Clutter/Noise",
Proc, item2,
Data, 1,
Foreground, "black",
NULL);

text90 = NRLX (Button, popup16, xPos, 400, yPos, 215,
Label, "2 - Signal/Noise",
Proc, item2,
Data, 2,
Foreground, "black",
NULL);

text91 = NRLX (Button, popup16, xPos, xcount, yPos, 240,
Label, "4 - Signal/(Clutter+Noise)",
Proc, item2,
Data, 4,
Foreground, "black",
NULL);

text92 = NRLX (Button, popup16, xPos, 300, yPos, 240,
Label, "8 - Prob. of Detection",
Proc, item2,
Data, 8,
Foreground, "black",
NULL);

text247 = NRLX (TextEntry, popup16, xPos, xcount, yPos, yycount,
Label, "Item 3",
Width, 50,
Height, 25,
PROC, sto13,
NULL);

text248 = NRLX (TextEntry, popup16, xPos, xcount, yPos, yycount=yycount+inc,
Label, "Item 4",
Width, 50,
Height, 25,
PROC, sto14,
NULL);

text93 = NRLX (TextEntry, popup16, xPos, xcount, yPos, yycount=yycount+inc,
Label, "Range Scale (nm/in.)",
Width, 550,
Height, 25,
PROC, sto14,
NULL);

text94 = NRLX (TextEntry, popup16, xPos, xcount, yPos, yycount=yycount+inc),

```

```

Label, "dB or Pd scale (dB/in. or unit/in.)",
width, 550,
height, 25,
proc, stof2,
NULL);

text95 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Maximum dB or Pd (dB or unit)",
width, 550,
height, 25,
proc, stof2,
NULL);

text96 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Maximum Z Dimension (in.)",
width, 550,
height, 25,
proc, stof4,
NULL);

text97 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Height Scale (ft/in.)",
width, 550,
height, 25,
proc, stof5,
NULL);

text98 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Fixed XMAX for Plotting (nmi)",
width, 550,
height, 25,
proc, stof6,
NULL);

/* read in an existing file, make changes on card 16 */
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v1
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno != 16) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v1
1, &v2, &v3, &v4, &v5, &v6);
        loc = ftell (fp1);
        printf ("win 16 1st loc = %d\n", loc);
        loc = loc + 80;
        print ("win 16 2nd loc=%d\n", loc);
        sprintf (&i1, "%d", i1);
        sprintf (&i2, "%d", i2);
        sprintf (&i3, "%d", i3);
        sprintf (&i4, "%d", i4);
        sprintf (&v1, "%f", v1);
        sprintf (&v2, "%f", v2);
        sprintf (&v3, "%f", v3);
        sprintf (&v4, "%f", v4);
        sprintf (&v5, "%f", v5);
        sprintf (&v6, "%f", v6);
        NRLXLoadTextEntry (text247, si3);
        NRLXLoadTextEntry (text248, si4);
        NRLXLoadTextEntry (text249, sv1);
        NRLXLoadTextEntry (text250, sv2);
        NRLXLoadTextEntry (text251, sv3);
        NRLXLoadTextEntry (text252, sv4);
        NRLXLoadTextEntry (text253, sv5);
        NRLXLoadTextEntry (text254, sv6);
    }
}
/* Popup Window 17 - Antenna Pattern Print/Plot Parameters */
void antpat()
{
    ycount=5, inc=70, xcount=10;
    cardno=17;
    i1=i2;i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;
    popup17 = NRLX (Popup, console, XPos, 530, YPos, 10,
Label, "Antenna Pattern Print/Plot Parameters",
width, 600,
height, 800,
background, "white",
framecolor, "blue violet",
NULL);
    text100 = NRLX (TextEntry, popup17, XPos, xcount, YPos, ycount,
label, "No Print or Plot, 1-Print, 2-Plot, 3-Print and Plot",
width, 550,
height, 25,
proc, stoll,
NULL);
    text101 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
label, "0 - Linear Scale, 1 - Polar Scale",
width, 550,
height, 25,
proc, stoll,
NULL);
    text249 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
label, "Item 3",
width, 550,
height, 25,
proc, stoll,
NULL);
    text250 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
label, "Item 4",
width, 550,
height, 25,
proc, stoll,
NULL);
}

```

```

text102 = NRLx (TextEntry, popup17, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Minimum Phi (deg.)",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

text103 = NRLx (TextEntry, popup17, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Maximum Phi (deg.)",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text104 = NRLx (TextEntry, popup17, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Delta Phi for Print & Plot (deg.)",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text105 = NRLx (TextEntry, popup17, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Scale Phi (deg./in.)",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text106 = NRLx (TextEntry, popup17, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Vertical Scale (dB/in.)",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text107 = NRLx (TextEntry, popup17, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Plot Height (in.)",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 17 */

if (cmdt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v1,
&v2, &v3, &v4, &v5, &v6);
    while (cardno != 17) {
        fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v1,
&v2, &v3, &v4, &v5, &v6);
        loc = ftell (fp1);
        printf ("win 17 1st loc = %d\n", loc);
        loc = loc - 80;
        printf ("win 17 2nd loc=%d\n", loc);
    }
}
loc = ftell (fp1);
printf ("win 17 loc = %d\n", loc);
loc = loc - 80;
printf ("win 17 loc=%d\n", loc);

sprint (si1, "%d", i1);
sprint (si2, "%d", i2);
sprint (si3, "%d", i3);
sprint (si4, "%d", i4);
sprint (sv1, "%f", v1);
sprint (sv2, "%f", v2);
sprint (sv3, "%f", v3);
sprint (sv4, "%f", v4);
sprint (sv5, "%f", v5);
sprint (sv6, "%f", v6);

NRLxLoadTextEntry (text100, s11);
NRLxLoadTextEntry (text101, s12);
NRLxLoadTextEntry (text102, s13);
NRLxLoadTextEntry (text103, s14);
NRLxLoadTextEntry (text104, s15);
NRLxLoadTextEntry (text105, s16);
NRLxLoadTextEntry (text106, s17);
NRLxLoadTextEntry (text107, s18);

/* popup window 18 - Atmospheric Attenuation Parameters */

void atmattn()
{
    ycount=5, inc=70, xcount=10;
    cardno=18;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;
}

popup18 = NRLx (POPUP, console, xpos, 530, ypos, 10,
Label, "Atmospheric Attenuation Parameters",
Width, 600,
Height, 800,
Background, "white",
Framecolor, "lime green",
NULL);

text110 = NRLx (TextEntry, popup18, xpos, xcount, ypos, ycount,
Label, "0-No Attenuation, 1-Linear Attenuation, 2-Asymptotic attenu-
nation",
text111 = NRLx (TextEntry, popup18, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "0 - Reset, # of Element",
Width, 550,
Height, 25,
Proc, stoi1,
NULL);

text112 = NRLx (TextEntry, popup18, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "variable NPOLY",
NULL);

```

```

Width, 550,
Height, 25,
Proc, sto13,
NULL);

text251 = NRLX (TextEntry, popup18, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, sto14,
NULL);

text113 = NRLX (TextEntry, popup18, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Attenuation value:(linear-db/nmi) or (Asym. -dB/angle)",
Width, 550,
Height, 25,
Proc, sto15,
NULL);

text114 = NRLX (TextEntry, popup18, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "variable A",
Width, 550,
Height, 25,
Proc, sto16,
NULL);

text115 = NRLX (TextEntry, popup18, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "variable B",
Width, 550,
Height, 25,
Proc, sto17,
NULL);

text116 = NRLX (TextEntry, popup18, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "variable R",
Width, 550,
Height, 25,
Proc, sto18,
NULL);

text117 = NRLX (TextEntry, popup18, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "variable K",
Width, 550,
Height, 25,
Proc, sto19,
NULL);

text118 = NRLX (TextEntry, popup18, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "variable 4",
Width, 550,
Height, 25,
Proc, sto20,
NULL);

/* read in an existing file, make changes on card 18 */
if (cndt == 1) {
    rewind (fp1);
}

1, &v2, &v3, &v4, &v5, &v6);
fscanf (fp1, "dddddfffff", &cardno, &i1, &i2, &i3, &i4, &v
while (cardno !=18) {
    fseek (fp1, 1, 1);
    fscanf (fp1, "dddddfffff", &cardno, &i1, &i2, &i3, &i4, &v
loc = ftell (fp1);
printf ("win 18 1st loc = %d\n", loc);
printf ("win 18 2nd loc=%d\n", loc);
sprint (sil, "id", i1);
sprint (s12, "id", i2);
sprint (s13, "id", i3);
sprint (s14, "id", i4);
sprint (sv1, "%f", v1);
sprint (sv2, "%f", v2);
sprint (sv3, "%f", v3);
sprint (sv4, "%f", v4);
sprint (sv5, "%f", v5);
sprint (sv6, "%f", v6);
NRLXLoadTextEntry (text10, sil);
NRLXLoadTextEntry (text11, s12);
NRLXLoadTextEntry (text12, s13);
NRLXLoadTextEntry (text13, s14);
NRLXLoadTextEntry (text14, sv1);
NRLXLoadTextEntry (text15, sv2);
NRLXLoadTextEntry (text16, sv3);
NRLXLoadTextEntry (text17, sv4);
NRLXLoadTextEntry (text18, sv5);
NRLXLoadTextEntry (text19, sv6);
/* Popup Window 19 - Rain Attenuation Parameters */
void rainatt()
{
    ycount=5, inc=70, xcount=10;
    cardno=19;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;
    popup19 = NRLX (Popup, console, XPos, 530, YPos, 10,
Label, "Rain Attenuation Parameters",
Width, 600,
Height, 800,
Background, "white",
Framecolor, "orange",
NULL);
text120 = NRLX (TextEntry, popup19, XPos, xcount, YPos, ycount,
Label, "0 - None, 1 - Rain",
Width, 550,
Height, 25,
Proc, sto19,
NULL);
}

```

```

        Label, "var 6",
        Width, 550,
        Height, 25,
        Proc, stof6,
        NULL);

text252 = NRLX (TextEntry, popup19, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Item 2",
Width, 550,
Height, 25,
Proc, sto2,
NULL);

text253 = NRLX (TextEntry, popup19, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
Proc, sto3,
NULL);

text254 = NRLX (TextEntry, popup19, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, sto4,
NULL);

text121 = NRLX (TextEntry, popup19, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Attenuation (dB/nmi)",
Width, 550,
Height, 25,
Proc, sto1,
NULL);

text122 = NRLX (TextEntry, popup19, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Minimum Range (nmi)",
Width, 550,
Height, 25,
Proc, sto2,
NULL);

text123 = NRLX (TextEntry, popup19, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Maximum Range (nmi)",
Width, 550,
Height, 25,
Proc, sto3,
NULL);

text255 = NRLX (TextEntry, popup19, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "var 4",
Width, 550,
Height, 25,
Proc, sto4,
NULL);

text256 = NRLX (TextEntry, popup19, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "var 5",
Width, 550,
Height, 25,
Proc, sto5,
NULL);

text257 = NRLX (TextEntry, popup19, xPos, xcount, yPos, (ycount=ycount+inc),
NULL);

/* popup window 20 - Jammer Parameters */
void jammer()
{
    ycount=5, inc=70, xcount=10;
    cardno=20;
    il=i2=13=i4=0;
    v1=v2=v3=v4=v5=v6=0;

popup20 = NRLX (Popup, console, xPos, 530, yPos, 10,
Label, "Jammer Parameters",
Width, 600,
Height, 800,
Background, "white",
FrameColor, "yellow",
NULL);

text258 = NRLX (TextEntry, popup20, xPos, xcount, yPos, ycount,
Label, "Item 1",
Width, 550,
Height, 25,
Proc, sto1,
NULL);

text259 = NRLX (TextEntry, popup20, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Item 2",
Width, 550,
Height, 25,
Proc, sto2,
NULL);

text260 = NRLX (TextEntry, popup20, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
Proc, sto3,
NULL);

text261 = NRLX (TextEntry, popup20, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, sto4,
NULL);

text130 = NRLX (TextEntry, popup20, xPos, xcount, yPos, (ycount=ycount+inc),
Label, "Jammer Effective Radiated Power (W)",
Width, 550,
NULL);
}

```

```

    Height, 25,
    Proc, stof1,
    NULL);

text131 = NRLX (TextEntry, popup20, xpos, xcount, ypos, (ycount=ycount+inc),
    Label, "Jammer Bandwidth (MHz)",
    Width, 550,
    Height, 25,
    Proc, stof2,
    NULL);

text132 = NRLX (TextEntry, popup20, xpos, xcount, ypos, (ycount=ycount+inc),
    Label, "Antenna Range (nmi)",
    Width, 550,
    Height, 25,
    Proc, stof3,
    NULL);

text133 = NRLX (TextEntry, popup20, xpos, xcount, ypos, (ycount=ycount+inc),
    Label, "Antenna Sidelobe Gain (dB)",
    Width, 550,
    Height, 25,
    Proc, stof4,
    NULL);

text134 = NRLX (TextEntry, popup20, xpos, xcount, ypos, (ycount=ycount+inc),
    Label, "SIC Cancellation Ratio (dB)",
    Width, 550,
    Height, 25,
    Proc, stof5,
    NULL);

text262 = NRLX (TextEntry, popup20, xpos, xcount, ypos, (ycount=ycount+inc),
    Label, "Var 6",
    Width, 550,
    Height, 25,
    Proc, stof6,
    NULL);

/* read in an existing file, make changes on card 20 */
if (ondt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno != 20) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
        loc = ftell (fp1);
        printf ("win 20 1st loc = %d\n", loc);
        loc = loc + 80;
        print ("win 20 2nd loc=%d\n", loc);
        sprint (&i1, "%d", i1);
        sprint (&i2, "%d", i2);
    }
}

loc = ftell (fp1);
printf ("win 20 1st loc = %d\n", loc);
loc = loc + 80;
print ("win 20 2nd loc=%d\n", loc);
sprint (&i1, "%d", i1);
sprint (&i2, "%d", i2);

```

```

    sprintf (&i3, "%d", i3);
    sprintf (&i4, "%d", i4);
    sprintf (&sv1, "%f", v1);
    sprintf (&sv2, "%f", v2);
    sprintf (&sv3, "%f", v3);
    sprintf (&sv4, "%f", v4);
    sprintf (&sv5, "%f", v5);
    sprintf (&sv6, "%f", v6);

NRLXloadTextEntry (text258, s11);
NRLXloadTextEntry (text259, s12);
NRLXloadTextEntry (text260, s13);
NRLXloadTextEntry (text261, s14);
NRLXloadTextEntry (text130, sv1);
NRLXloadTextEntry (text131, sv2);
NRLXloadTextEntry (text132, sv3);
NRLXloadTextEntry (text133, sv4);
NRLXloadTextEntry (text134, sv5);

/* popup window 1 - Execute */
void execute()
{
    ycount=5, inc=70, xcount=10;
    cardno=1;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;
    popup1 = NRLX (Popup, console, xPos, 530, yPos, 10,
                    Label, "Execute",
                    width, 600,
                    height, 800,
                    background, "white",
                    framecolor, "dark turquoise",
                    NULL);
    text140 = NRLX (TextEntry, popup1, xPos, xcount, yPos, ycount,
                    Label, "0 - Clutter + Noise, 1 - Noise Only",
                    width, 550,
                    height, 25,
                    proc, stoi1,
                    NULL);
    text141 = NRLX (TextEntry, popup1, xPos, xcount, yPos, (ycount=ycount+inc),
                    Label, "0 - No pattern, print, plot, 1-print, plot, pattern, continue",
                    width, 550,
                    height, 25,
                    proc, stoi2,
                    NULL);
    text142 = NRLX (TextEntry, popup1, xPos, xcount, yPos, (ycount=ycount+inc),
                    Label, "0 - Print, 1 - No Print",
                    width, 550,
                    height, 25,
                    proc, stoi3,
                    NULL);
}

```

```

text263 = NRLX (TextEntry, popup1, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, stoi4,
NULL);

text143 = NRLX (TxtEntry, popup1, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Single Target Altitude (ft.)",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

text264 = NRLX (TextEntry, popup1, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 2",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text265 = NRLX (TextEntry, popup1, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 3",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text266 = NRLX (TextEntry, popup1, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text267 = NRLX (TextEntry, popup1, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 5",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text268 = NRLX (TextEntry, popup1, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 6",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 1 */

if (cmdt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%ff%ff", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno != -1) {
        loc = ftell (fp1);
        printf ("win 1\n", loc);
        loc = loc + 80;
        printf ("win 1 2nd loc=%d\n", loc);
        sprintf (&s1, "%d", i1);
        sprintf (&s2, "%d", i2);
        sprintf (&s3, "%d", i3);
        sprintf (&s4, "%d", i4);
        sprintf (&sv1, "%f", v1);
        sprintf (&sv2, "%f", v2);
        sprintf (&sv3, "%f", v3);
        sprintf (&sv4, "%f", v4);
        sprintf (&sv5, "%f", v5);
        sprintf (&sv6, "%f", v6);
        NRLXLoadTextEntry (text140, s1);
        NRLXLoadTextEntry (text141, s2);
        NRLXLoadTextEntry (text142, s3);
        NRLXLoadTextEntry (text143, s4);
        NRLXLoadTextEntry (text263, sv1);
        NRLXLoadTextEntry (text264, sv2);
        NRLXLoadTextEntry (text265, sv3);
        NRLXLoadTextEntry (text266, sv4);
        NRLXLoadTextEntry (text267, sv5);
        NRLXLoadTextEntry (text268, sv6);
    }
}

/* Popup Window 11 - Title for printout and plot */
void title()
{
    cardno=11;
    ycount=5, xcount=10, inc=70;
    i1=i2=13=i4=0;
    v1=v2=v3=v4=v5=v6=0;
    popup11 = NRLX (popup, console, xpos, 530, ypos, 20,
Label, "Title for Printout and Plot",
Width, 600,
Height, 800,
Background, "white",
FrameColor, "dark green",
NULL);
}

text150 = NRLX (TextEntry, popup11, xpos, xcount, ypos, ycount,
Label, "# of Title Line to Follow (maximum is 4)",
Width, 580,
Height, 25,
Proc, stoi1,
NULL);

```

```

NRLXLoadTextEntry (text151, rstr1);
NRLXLoadTextEntry (text152, rstr2);

text151 = NRLX (TextEntry, popup11, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Line one",
Width, 580,
Height, 25,
Proc, string1,
NULL);

text152 = NRLX (TextEntry, popup11, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Line two",
Width, 580,
Height, 25,
Proc, string2,
NULL);

text153 = NRLX (TextEntry, popup11, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Line three",
Width, 580,
Height, 25,
Proc, string3,
NULL);

text154 = NRLX (TextEntry, popup11, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Line four",
Width, 580,
Height, 25,
Proc, string4,
NULL);

/* read in an existing file, read a line and check for cardno 11 */

if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v1,
&v2, &v3, &v4, &v5, &v6);
}

while (cardno != 11) {
    fscanf (fp1, "%d", &cardno);
    fscanf (fp1, "%d", &i1);
    printf ("card 11 - read in loc = %d\n", loc);
    sprintf (s1, "%d", i1);
    NRLXLoadTextEntry (text150, s1);
    printf ("card 11 - read in loc2 = %d\n", loc2);
}

if (i1 == 1) {
    fseek (fp1, loc2, 0);
    gets (rstr1, 78, fp1);
    NRLXLoadTextEntry (text151, rstr1);
    else if (i1 == 2) {
        fseek (fp1, loc2, 0);
        gets (rstr1, 78, fp1);
        fseek (rstr1, loc2+1, 0);
        gets (rstr2, 78, fp1);
        fseek (fp1, loc2+162, 0);
        gets (rstr3, 78, fp1);
        NRLXLoadTextEntry (text151, rstr2);
        NRLXLoadTextEntry (text152, rstr3);
    }
}

else if (i1 == 3) {
    fseek (fp1, loc2, 0);
    gets (rstr1, 78, fp1);
    fseek (fp1, loc2+81, 0);
    gets (rstr2, 78, fp1);
    fseek (fp1, loc2+162, 0);
    gets (rstr3, 78, fp1);
    NRLXLoadTextEntry (text151, rstr2);
    NRLXLoadTextEntry (text153, rstr3);
}

else if (i1 == 4) {
    fseek (fp1, loc2, 0);
    gets (rstr1, 78, fp1);
    fseek (fp1, loc2+81, 0);
    gets (rstr2, 78, fp1);
    fseek (fp1, loc2+162, 0);
    gets (rstr3, 78, fp1);
    fseek (fp1, loc2+243, 0);
    gets (rstr4, 78, fp1);

    NRLXLoadTextEntry (text151, rstr1);
    NRLXLoadTextEntry (text152, rstr2);
    NRLXLoadTextEntry (text153, rstr3);
    NRLXLoadTextEntry (text154, rstr4);
}

/* title for printout and plot */
void string1 (string lnl)
{
    printf ("%s", lnl);
    printf ("C - line one : %s\n", lnl);
    if (cndt == 1) lne1=l;
}

void string2 (string ln2)
{
    sprintf (str2, "%s", ln2);
    printf ("%s - str1= %s\n", str1);
    sprintf (astr1, "%s", str1);
    if (cndt == 1) lne2=1;
}

```

```

void string3 (String ln3)
{
    sprintf(str3, "%s", ln3);
    printf ("string3 - str3= %s\n",str3);
    sprintf (astr3,"%-78s", str3);

    if (cndt == 1) line3=1;
}

void string4 (String ln4)
{
    sprintf(str4, "%s", ln4);
    printf ("string4 - str4= %s\n",str4);
    sprintf (astr4,"%-78s", str4);

    if (cndt == 1) line4=1;
}

/* item 1 on card 16 */
void item16 (long data)
{
    printf("item 1 on card 16: %d\n", data); /*/
    i1=data;
}

/* item 2 on card 16 */
void item2 (long data)
{
    printf("item 2 on card 16: %d\n", data); /*/
    i2=data;
}

/* string to integer */
void sttoi (String var1)
{
    printf("Incoming from text widget: %s\n",var1); /*/
    sscanf (var1, "%d", &i1);
}

/* Result of conversion: %f\n",v1); /*/
void stof1 (String var5)
{
    printf("Incoming from text widget: %s\n",var5); /*/
    /* MUST use ADDRESS of numeric variable---- &v1 -----*/
    sscanf (var5, "%f", &v1);
}

/* string to floating point */
void stof2 (String var6)
{
    sscanf (var6, "%f", &v2);
}

/* stof3 (String var7)
void stof3 (String var7)
{
    sscanf (var7, "%f", &v3);
}

/* stof4 (String var8)
void stof4 (String var8)
{
    sscanf (var8, "%f", &v4);
}

/* stof5 (String var9)
void stof5 (String var9)
{
    sscanf (var9, "%f", &v5);
}

/* stof6 (String var10)
void stof6 (String var10)
{
    sscanf (var10, "%f", &v6);
}

/* Open a new data file */
void openfile()
{
    cndt=2;
}

/* get the filename */
printf ("%s File name = ");
gets (fname);
if ((fp = fopen(fname, "w")) == NULL)
{
    printf(" Unable to open %s\n", fname);
    exit(1);
}

```

```

    }

    /* Close the data file */
    void closeinfile()
    {
        cardno=0;
        i1=i2=3-i4=0;
        v1-v2=v3-v4=v5-v6=0;

        fprintf (fp, "#2d#4d#6d#4d#10.4f#10.4f#10.4f#10.4f\n", ca
rdno,i1,i2,i3,i4,v1,v2,v3,v4,v5,v6);

        fclose(fp);
    }

    /* save the window */
    void savewin()
    {
        if (cardno == 11) {
            fprintf (fp, "#2d#4d#6d#4d#10.4f#10.4f#10.4f\n", 
cardno,i1,i2,i3,i4,v1,v2,v3,v4,v5,v6);

            if (cardno == 11) {
                fprintf (fp, "#2d#4d#6d#4d#10.4f#10.4f#10.4f\n", c
ardno,i1,i2,i3,i4,v1,v2,v3,v4,v5,v6);

                if (i1 == 1) fprintf (fp, "#2d#-78s\n", cardno, astr1);
                else if (i1 == 2) fprintf (fp, "#2d#-78s\n2d#-78s\n", cardno, astr1,
astr2);
                else if (i1 == 3) fprintf (fp, "#2d#-78s\nn2d#-78s\nn2d#-78s\n", card
o, astr1, cardno, astr2);
                else if (i1==4) fprintf (fp, "#2d#-78s\nn2d#-78s\nn2d#-78s\n", 
cardno, astr1, cardno, astr2, cardno, astr4);
            }
        }
    }

    /* open an existing data file */
    void openinfile()
    {
        cndt=1;
        printf (" File name for read = ");
        gets (infilename);
        if ((fp1 = fopen(infilename, "r+")) == NULL)
            printf(" Unable to open %s\n", infilename);
        exit(1);
    }
}

```

```
    fputs (astr4, fp1);
}

else if (cardno != 11) {
    fseek (fp1, loc, 0);
    fprintf (fp1, "%2dt%4d%6d%4d%4d%10.4f%10.4f%10.4f%10.4f\n",
cardno,i1,i2,i3,i4,v1,v2,v3,v4,v5,v6);
}

/* Quit function */
void quit()
{
    exit(0);
}
```